TESIS

# MEMBANGUN HUBUNGAN KERUNUTAN ARTIFAK PADA LINGKUNGAN PENGEMBANGAN CEPAT

HENGKI SUHARTOYO
5111201030

DOSEN PEMBIMBING
Dr. Ir. Siti Rochimah, MT.

PROGRAM MAGISTER
BIDANG KEAHLIAN REKAYASA PERANGKAT LUNAK
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

THESIS

# AUTOMATIC TRACEABILITY LINK ON AGILE SOFTWARE DEVELOPMENT

HENGKI SUHARTOYO
5111201030

SUPERVISOR
Dr. Ir. Siti Rochimah, MT.

MASTER PROGRAM
SOFTWARE ENGINEERING
INFORMATICS ENGINEERING
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

# LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
HENGKI SUHARTOYO
NRP. 5111201030

Dengan Judul :
Membangun Hubungan Kerunutan Artifak pada Lingkungan Pengembangan Cepat

Tanggal Ujian : 24 Juni 2016
Periode Wisuda : 2016 Genap

Disetujui oleh :

1. Dr. Ir. Siti Rochimah, MT
   NIP. 196810021994032001 (Pembimbing)
2. Daniel Oranova, S.Kom. MSc. PD.Eng.
   NIP. 197411232006041001 (Penguji)
3. Umi Laili Yuhana, S.Kom, M.Sc
   NIP. 197906262005012002 (Penguji)
4. Rizky Januar Akbar, S.Kom., M.Eng.
   NIP. 198701032014041001 (Penguji)

Direktur Program Pasca Sarjana,

Prof.Ir.Djauhar Manfaat, M.Sc., Ph.D.
NIP. 196012021987011001

MEMBANGUN HUBUNGAN KERUNUTAN ARTIFAK PADA LINGKUNGAN PENGEMBANGAN CEPAT

Name : Hengki Suhartoyo
NRP : 5111201030
Pembimbing : Dr. Ir. Siti Rochimah, MT.

# ABSTRAK

Kerunutan merupakan mekanisme penting untuk mengelola dan mengaudit proses pengembangan perangkat lunak, dapat dikatakan bahwa semua artefak dari pengembangan perangkat lunak diarahkan dan berkaitan dengan kebutuhan perangkat lunak. Kerunutan digunakan untuk berbagai tujuan, termasuk untuk manajemen kebutuhan, manajemen perubahan, dampak analisis, verifikasi, validasi, dan audit. Untuk membantu proses membangun kerunutan hubungan antar artefak perlu dikembangkan alat bantu yang dapat secara otomatis menghubungkan antar artefak didalam lingkungan pengembangan cepat. Beberapa artefak utama dalam lingkungan pengembangan cepat adalah cerita pengguna dan kode sumber. Pada penelitian ini dikembangkan kakas bantu untuk membentuk hubungan kerunutan antara kedua artefak tersebut yang berbahasa Indonesia.

Pada penelitian ini diusulkan metode yang dapat membentuk hubungan kerunutan secara otomatis menggunakan metode pencocokan string antara cerita pengguna dengan kode sumber. Cerita pengguna dan kode sumber diekstraksi menjadi himpunan kata dasar dan dilakukan pencocokan kata antara keduanya menggunakan algoritma trigram. Ekstraksi cerita pengguna menggunakan algoritma *Nazief & Adriani*. Ekstraksi kode sumber menggunakan pustaka *javaparser* dan dioptimasi menggunakan pendekatan konvensi penamaan java. Hasil dari pencocokan dioptimalkan dengan memberikan ambang batas jumlah kata yang terkandung dalam kode sumber dibandingkan dengan jumlah kata yang terdapat pada cerita pengguna.

Sistem diuji dengan tiga dataset cerita pengguna beserta kode sumbernya yaitu *smartPortal* terdiri dari 17 cerita pengguna dan 52 file kode sumber, *smartAbsensi* terdiri dari 47 cerita pengguna dan 161 file kode sumber yang ketiga *smartTravel* 80 cerita pengguna dan 222 file kode sumber. Hasil otomatisasi hubungan kerunutan oleh sistem dibandingkan dengan hasil kerunutan hubungan oleh pengembang diperoleh nilai presisi *smartPortal* 0.288, *smartAbsensi* 0.285 dan smartTravel 0.213. Rata-rata hasil presisi dibawah 0.5 disebabkan dataset masih banyak menggunakan identifier berbahasa inggris dan singkatan-singkatan.

Kata kunci: *Kerunutan, lingkungan pengembangan cepat, cerita pengguna pencocokan string.*

AUTOMATIC TRACEABILITY LINK ON AGILE SOFTWARE DEVELOPMENT

Name : Hengki Suhartoyo
Student Identity Number : 5111201030
Supervisor : Dr. Ir. Siti Rochimah, MT.

# ABSTRACT

Traceability is an important mechanism to manage and audit the software development process, all of the artifacts of software development aimed and related to the software requierement. Kerunutan is used for various purposes, including to requirement management, change management, impact analysis, verification, validation, and audit. To help the process of establishing traceability links between artifacts is necessary to develop automatic traceability in the rapid application development environment. Some of the major artifacts in a rapid development environment (agile) are user story and source code.

This research proposed a method to establish traceability link between user story and java source code in Indonesian Language. User story and source code are extracted into a set of basic word, then matched both of them using trigram algorithm. User story extraction using *Nazief & Adriani algorithm*. Extraction of the source code using the library javaparser and optimized by java naming convention approaches. The results of matching words are optimized by providing a threshold number of words contained in the source code as compared to the number of words contained in the story.

The automatic traceabillty tool have been made tested using three dataset that containing user story and java source code. First dataset is *smartPortal* containing 17 user stories and 52 files source code. Second dataset is *smartAbsensi* containing 47 user stories and 52 files source codes. Third dataset is *smartTravel* that containing 80 user stories and 222 files souce codes. The results of automatization traceabillity link compared with traceability link by developer. The result of comparation ranked using precision recall with the output are: smartPortal 0.288, smartAbsensi 0.285 and smartTravel 0.213. The average results of the precision of less than 0.5 caused datasheet still using the English language identifier and abbreviations.

Keywords: *traceabillity, agile, user story , string matching.*

# DAFTAR ISI

# DAFTAR GAMBAR

# DAFTAR TABEL

# BAB I
# PENDAHULUAN

## 1.1.Latar Belakang

Kerunutan (*Traceability*) merupakan mekanisme penting untuk mengelola dan mengaudit proses pengembangan perangkat lunak (P. Lago, 2009), dapat dikatakan bahwa semua artefak dari pengembangan perangkat lunak diarahkan dan berkaitan dengan kebutuhan perangkat lunak. Kerunutan digunakan untuk berbagai tujuan, termasuk untuk manajemen kebutuhan, manajemen perubahan, dampak analisis, verifikasi, validasi, dan audit. Namun dalam lingkungan pengembangan perangkat lunak cepat terdapat berbagai macam artefak yang terdapat dalam satu dokumentasi, hal ini mencerminkan fakta bahwa dalam model pengembangan perangkat lunak cepat setiap artefak dapat digunakan sebagai dokumentasi. (Matthias, 2013) artefak penting dalam melakukan pemodelan perangkat lunak diantaranya yaitu Diagram Kasus Pengguna, Diagram Kelas, *Diagram urutan* , Diagram Kolaborasi, Ujicoba Kasus, dan kode sumber. Artefak-artefak tersebut saling berhubungan satu dengan yang lainnya. Mengetahui kerunutan antar artefak tersebut sangat dibutuhkan dalam perancangan perangkat lunak. Dengan banyaknya artefak yang akan ditelusuri kerunutan hubungannya maka akan semakin memerlukan waktu dan biaya.

Artefak didalam pengembangan perangkat lunak tangkas (*agile software development)* selanjutnya disebut *Agile* berbeda dengan metodologi pengembangan perangkat lunak seperti dengan metodologi maupun Metodologi Berbasis Obyek (*Object Oriented Methodologies*) (Kenneth , 2011). Agile sebagai metodologi pengembangan system alternative memiliki beberapa jenis diantaranya yang paling popular yaitu: *Scrum, eXtreme Programming(XP), Kanban, Feature Driven Development (FDD) dan Agile Unified Process (AUP).* Dari berbagai jenis tersebut agile menimbulkan berbagai macam istilah dan artefak yang belum pernah diklasifikasikan sebelumnya, sehingga *Matthias GrÖber* melakukan penelitian pada 76 paper yang membahas tentang agile, dan menemukan bahwa didalam 73 paper terdapat 314 nama artefak yang berbeda-beda, sehingga artefak pada agile diklasifikasikan menjadi: artefak final, spesifikasi kebutuhan, kasus uji, proses

produksi, manajemen proyek dan *Backlog Item* (Matthias, 2013). Pada agile tingkat perubahan artefak-artefak relative lebih sering terjadi karena antara pengguna dan pengembang terlibat menjadi satu kesatuan tim pengembangan system yang terlibat secara intensif, sehingga antara perubahan RS dan kode sumber dapat dipastikan berubah secara cepat.

Selama pembangunan, artefak perangkat lunak berubah terus-menerus itulah sebabnya metode sederhana untuk pemulihan hubungan menjadi tidak efisien. Suatu pendekatan yang memanfaatkan metode pemulihan kerunutan hubungan untuk menyediakan kerunutan hubungan terus berkembang. Otomatisasi dari kerunutan hubungan sebelumnya menggunakan algoritma *Latent Semantic Indexing* mengevaluasi kembali dari kerunutan hubungan yang terus mengalami perubahan. Metode yang digunakan menggunakan semua informasi yang tersedia sebelumnya dan hanya mengevaluasi dampak perubahan pada kerunutan tersebut dengan demikian biaya untuk pemulihan kerunutan akan dihindari untuk versi selanjutnya (Rilling, 2007).

Dalam banyak penelitian tentang kerunutan hubungan pengembangan perangkat lunak, sebagian besar menangani masalah pemulihan kerunutan hubungan untuk mengidentifikasi hubungan semantik antara kode sumber dan dokumentasinya pada metode pengembangan tradisional maupun OOM. Pada kesempatan ini penulis mencoba untuk menawarkan otomatisasi kerunutan hubungan didalam lingkungan pengembangan cepat antara cerita pengguna dengan kode sumber menggunakan pencocokan string. Penulis memilih hubungan kerunutan tersebut karena dalam lingkungan pengembangan cepat khususnya *scrum* perubahan iterasi atau sprint terjadi cukup cepat dari satu versi ke versi berikutnya, sehingga dibutuhkan alat yang dapat mengetahui dengan cepat hubungan kerunutan antara cerita pengguna dengan kode sumber. Dengan string matching, pencarian string yang mirip antara cerita pengguna dengan kode sumber diterapkan sebagai mekanisme penentuan kerunutan hubungan tanpa membatasi kebebasan dalam mengekspresikan kebutuhan dalam Cerita Pengguna . Hubungan diidentifikasi dengan mencari kecocokan antara kata-kata didalam cerita pengguna dengan kode sumber yang telah diekstraksi menjadi nama-nama class, method, atrribut, dan komentar.

## 1.2.Perumusan Masalah

Permasalahan yang akan diselesaikan dalam penelitian ini adalah:

1. Bagaimana melakukan ekstraksi pada cerita pengguna menjadi class, attribute, method dan penjelasan?
2. Bagaimana melakukan ekstraksi pada kode sumber *Java* menjadi class, attribute dan method?
3. Bagaimana menentukan hubungan kerunutan antara cerita pengguna dengan kode sumber.
4. Bagaimana mencari ketimpangan hubungan antara cerita pengguna dengan kode sumber.

## 1.3.Batasan Masalah

Untuk mendapatkan hasil sesuai dengan yang diharapkan, penelitian ini dibatasi pada hal-hal berikut:

a. Cerita pengguna yang digunakan menggunakan merupakan cerita pengguna pada lingkungan pengembangan cepat.
b. Kode sumber yang digunakan dalam penelitian ini adalah kode sumber bahasa pemrograman Java.
c. Bahasa yang digunakan dalam cerita Pengguna maupun penamaan attribute, class, dan method menggunakan bahasa Indonesia.

## 1.1. Tujuan dan Manfaat Penelitian

Penelitian ini bertujuan untuk membentuk hubungan kerunutan antara Cerita Pengguna dengan kode program menggunakan string matching secara otomatis.

Penelitian ini bermanfaat untuk membantu perawat perangkat lunak untuk menemukan ketimpangan hubungan kerunutan antara cerita pengguna dengan kode program selama proses evolusi perangkat lunak.

# BAB II
# KAJIAN PUSTAKA

## 2.1 Scrum

Scrum adalah sebuah kerangka kerja di mana orang-orang dapat menyelesaikan permasalahan kompleks yang senantiasa berubah, di mana pada saat bersamaan menghasilkan produk dengan nilai setinggi mungkin secara kreatif dan produktif.
Scrum bersifat:

- Ringan
- Mudah dipahami
- Sulit dikuasai

Scrum adalah kerangka kerja proses yang telah digunakan untuk mengelola pengembangan produk kompleks semenjak awal tahun 1990-an. Scrum bukanlah sebuah proses ataupun teknik untuk mengembangkan produk, daripada itu, ini adalah sebuah kerangka kerja di mana di dalamnya dapat dimasukkan beragam proses dan teknik. Scrum akan mengekspos pergerakan efektifitas manajemen produk dan praktik pengembangan yang sedang anda jalani, dengan begitu anda dapat melakukan peningkatan.

Kerangka kerja Scrum terdiri dari Tim Scrum, serta peran-peran mereka di dalamnya, acara-acara, artefak-artefak dan aturan-aturan. Setiap komponen di dalam kerangka kerja memiliki maksud tertentu dan peran penting demi keberhasilan penggunaan Scrum. Aturan main Scrum menyatukan acara-acara, peran-peran dan artefak-artefak, menjaga harmonisasi dan interaksi antar setiap komponen.

## 2.2 Kode Sumber

Kode sumber adalah suatu rangkaian peryataan atau deklarasi yang ditulis dalam bahasa pemrograman komputer yang terbaca manusia. Kode sumber yang menyusun suatu program biasanya disimpan dalam satu atau lebih berkas teks (Juergen , 2007). Kode sumber biasanya ditransformasikan (di-*compile*) oleh *compiler program* dari kode sumber menjadi *low level machine code* yang dapat dimengerti oleh computer atau menjadi executable file. Contoh kode program dalam bahasa pemrograman java adalah sebagai berikut:

```
class HelloWorld {
      public static void main(String[] args) {
      //Tampilkan Hello World pada terminal window
            System.out.println("Hello World!");
      }
}
```

Gambar 2.1 Contoh kode program java

2.3.Cerita Pengguna

Cerita Pengguna sebagai sebagai bagian Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Cerita Pengguna merupakan bentuk utama dari *requirement expression* (ekspresi kebutuhan) yang digunakan dalam pengembangan cepat (*Agile)*, SKPL akan menjadi kompilasi dari Cerita Pengguna dan SKPL berkembang sebagai suatu sistem yang berkembang.

Setiap Cerita Pengguna dalam agile merupakan pernyataan singkat yang digunakan untuk menguraikan apa yang sistem perlu lakukan untuk pengguna. Biasanya, setiap cerita pengguna dinyatakan dalam bentuk sebagai berikut:

As a <role> I can <activity> so that < value>

Dimana

- *Role* – menggambarkan *siapa* yang melakukan aksi
- *Activity* – Menggambarkan aksi yang dilakukan
- *Value* – Menggambarkan nilai yang dihasilkan dari aksi yang dilakukan

| “Sebagai Kasir Saya dapat Mencetak Laporan Setiap Shift sehingga saya mengetahui berapa uang kas, kartu kredit maupun kartu debit yang saya terima.” |
|---|

Gambar 2.2 Contoh cerita pengguna

Untuk menyatakan bahwa Cerita Pengguna telah sesuai dengan yang diinginkan atau dianggap selesai oleh user maka Cerita Pengguna harus dilengkapi dengan kriteria penerimaan (*Acceptance Criteria*). Kriteria penerimaan menentukan batasan dari cerita pengguna, serta mengkonfirmasi bahwa cerita pengguna telah lengkap, selesai dan dapat bekerja sebagaimana dimaksud.

Dari contoh diatas dapat dilengkapi dengan kriteria penerimaan sebagai berikut:

- Terdapat pilihan periode shift untuk setiap kasir
- Dapat mencetak rekap seluruh kasir setiap shift
- Laporan dapat secara otomatis menambahkan kolom jenis pembayaran

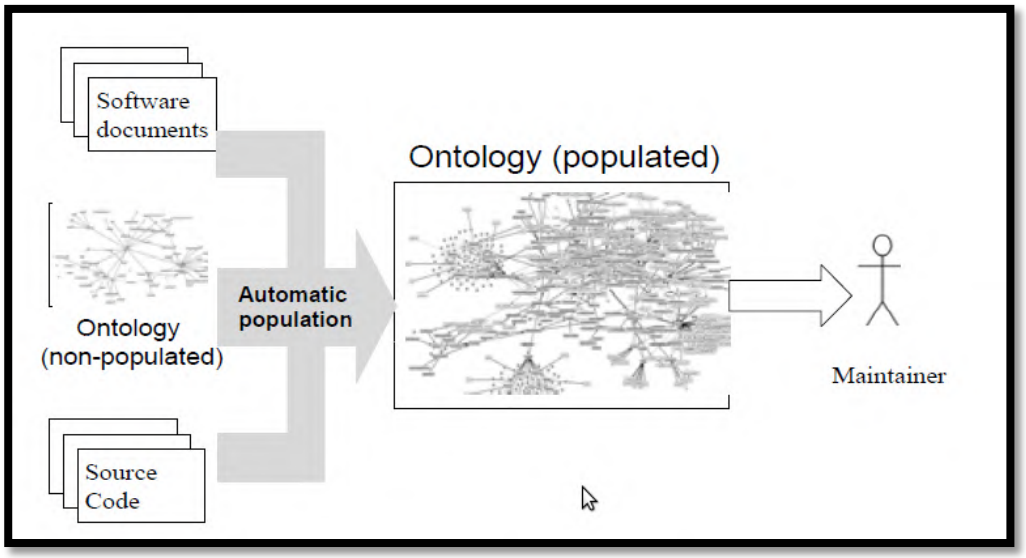
## 2.4.Model-Model Hubungan Kerunutan

Pemelihara perangkat lunak (*Softtware maintainers*) secara rutin selalu berhubungan dengan perangkat lunak artefak seperti kode sumber dan dokumentasi perangkat lunak. Seringkali hubungan antar artefak terputus karena perbedaan representasi dan tingkat abstraksi. Salah satu tantangannya adalah memulihkan (*recovery*) dan menjaga hubungan *semantic* antar artefak. Dalam penelitian ini menyajikan pendekatan baru untuk masalah kerunutan dengan menciptakan representasi formal ontological dari dokumentasi perangkat lunak dengan kode sumber. Representasi yang dihasilkan kemudian disesuaikan untuk membangun hubungan kerunutan pada tingkat semantik. Pertanyaan ontologis dan penalaran dapat diterapkan pada representasi ini untuk menyimpulkan dan menetapkan hubungan kerunutan tambahan untuk mendukung tugas-tugas perawatan yang spesifik.

Hubungan kerunutan membantu pembuat perangkat lunak memahami hubungan dan saling ketergantungan antar perangkat lunak artefak. Pada kenyataanya, artefak perangkat lunak yang dibuat sebagai bagian dari proses-proses ini akhirnya akan terputus dari satu sama lain. Dalam penelitian ini menyajikan pendekatan baru untuk masalah kerunutan dengan menciptakan representasi formal ontological dari dokumentasi perangkat lunak dengan kode sumber.

Representasi yang dihasilkan kemudian disesuaikan untuk membangun kerunutan hubungan pada tingkat semantik. *Juergen Rilling dkk* menggunakan Text Mining (TM) untuk menganalisis dokumentasi perangkat lunak dan parsing kode sumber untuk menganalisis kode sumber. (Juergen , 2007)

Sebagai gambaran umum dari pendekatan metodologi ini ditunjukkan pada Gambar 2.1. Yang pertama, ontologi yang sudah ada secara otomatis diisi dari kode sumber dan dokumentasi artefak. Pada tahap kedua, dasar pengetahuan yang dihasilkan dieksplorasi untuk membangun hubungan kerunutan guna memberikan dukungan bagi pengelola.

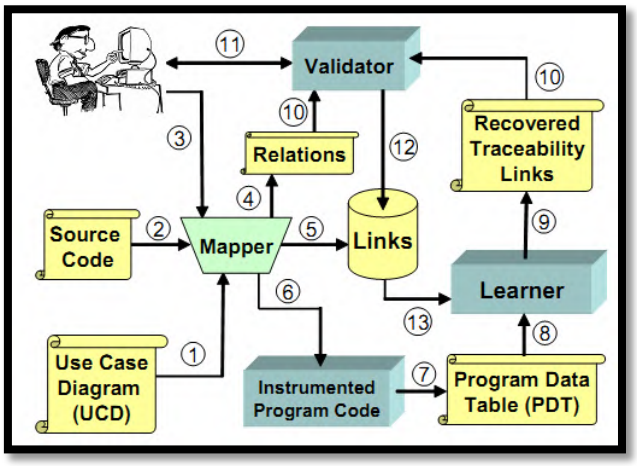



Gambar 2.3 Model .Kerunutan dengan pendekatan ontologi

Diagram Kasus Penggunas (UCDs) secara umum digunakan untuk menggambarkan kebutuhan dan fungsi yang diinginkan dari produk perangkat lunak. Namun UCDs berhubungan dengan kode sumber dan *maintaining traces* antara kode sumber dengan elemen UCDs ini membantu developer untuk memahami kode yang terus berkembang. Kontribusi dari paper ini ada dua. Pertama, ditawarkan pendekatan baru untuk mengotomatisasi bagian dari proses Kerunutan hubungan (TLs) antara jenis dan variabel dalam program Java dan unsur-unsur UCDs. *Prototipe* yang dihasilkan dievaluasi pada perangkat lunak kode terbuka dan komersial, dan hasilnya menunjukkan bahwa pendekatan yang digunakan dapat memulihkan TLs dengan otomatisasi tingkat tinggi dan presisi. Kedua, dikembangkan sebuah plugin Eclipse yang memungkinkan program untuk melacak jenis program dan variabel untuk unsur UCDs dan menggunakan TLs. Studi kasus yang dilakukan menunjukkan bahwa *developer* mengembangkan perangkat lunak lebih efisien dengan plugin yang dikembangkan. Developer mengembangkan TLs secara bersama-sama untuk mengurangi beban developer.

Otomatisasi dalam proses TLs untuk parameter jenis dan variabel dalam program java pada UCD. Pengujian dilakukan pada perangkat lunak kode terbuka dan komersial. Teknik yang digunakan dengan *LEarning and ANAlyzing Requirements Trace-ability* (LeanArt) sesuai Gambar 2.2 . Penerapan sistem ini di pengujian pada *Vehicle Maintenance Tracker* (VMT) project. Dengan menghubungkan kode sumber

dengan Diagram Kasus Pengguna maka menghasilkan pendekatan yang dilakukan dapat *Recovering traceabillity links* dengan sangat sesuai. (Mark , 2007)




Gambar 2.4 Model . LeanART architecture

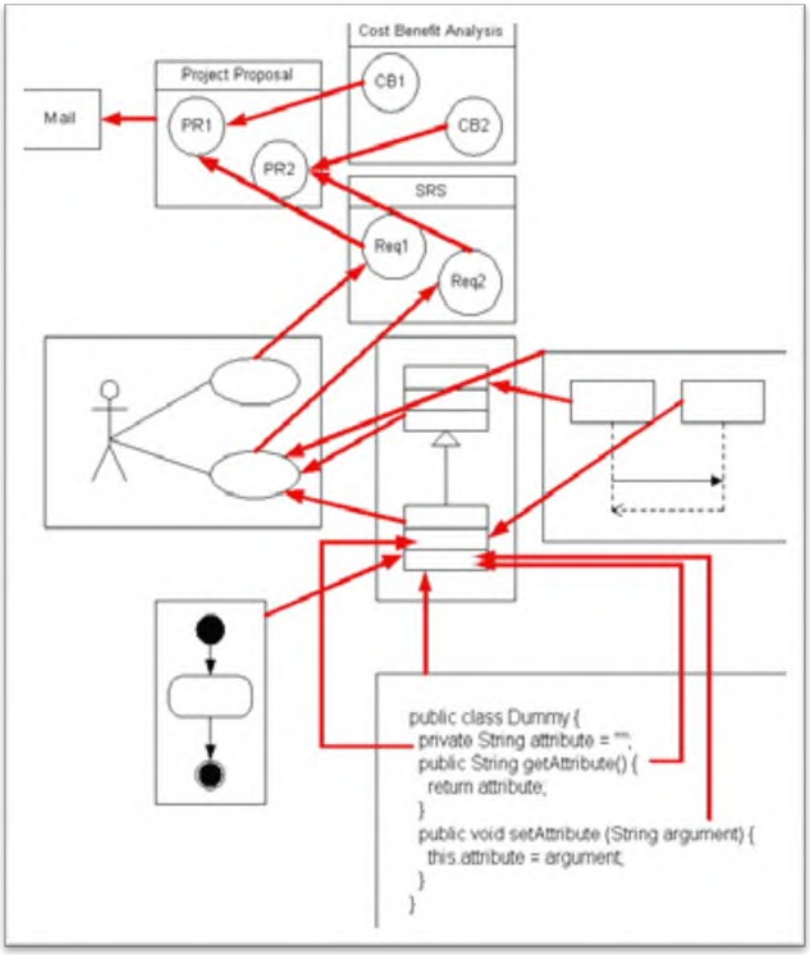
Tujuan utama dalam kerunutan perangkat lunak adalah untuk melacak semua elemen yang dapat dianggap cukup relevan bagi organisasi dalam proyek tertentu atau produk software. Beberapa contoh klasik dari unsur-unsur persyaratan, desain, kode sumber atau tes. Namun ada sejumlah informasi yang dianggap tidak hati-hati dalam literatur saat ini. Email yang dikirim oleh para pemangku kepentingan, risalah rapat, proposal proyek atau analisis biaya manfaat adalah dokumen yang juga merupakan bagian penting dari produk perangkat lunak, mempertahankan sejumlah besar pengetahuan yang diperlukan oleh organisasi (misalnya untuk mengelola perbaikan proses dan penentuan kemampuan) (Héctor , 2007)

Kemampuan untuk membangun dan memelihara hubungan antara unsur-unsur yang terkandung dalam dokumen ini dan lainnya sangat penting, tidak peduli tipologi atau tahap selama siklus hidup produk di mana mereka muncul. Mengelola semua jenis hubungan antara unsur-unsur apa saja yang adalah apa yang kita definisikan sebagai total kerunutan.

Pada gambar 2.3 menggambarkan konsep kerunutan total. Sebuah proyek perangkat lunak tidak mulai dalam tahap rekayasa persyaratan, seperti yang ditunjukkan oleh banyak penulis. Selain itu, kita tidak harus mempertimbangkan persyaratan sebagai inti dalam kerunutan, bahkan ketika perspektif yang lebih luas, termasuk informasi yang tidak secara langsung berkaitan dengan proses rekayasa seperti yang dijelaskan dalam.

*Hector dkk* mengklaim bahwa kerunutan perangkat lunak harus difokuskan dalam membangun hubungan antara unsur-unsur yang membentuk produk, terlepas dari jenis atau tahap di mana pertama kali muncul.

Gambar 2.3 menunjukkan kasus hipotetik sederhana,sehingga kita dapat dengan mudah menemukan di dunia nyata. Siklus hidup produk dimulai ketika pengguna mengirimkan email yang meminta untuk beberapa jenis pertemuan dalam kebutuhan perangkat lunak.




Gambar 2.5 Contoh dasar kerunutan secara total

*2.5.*Perbandingan Hubungan Kerunutan Kode Sumber

Dalam penelitian yang dilakukan oleh *Juergen Rilling* dkk, menyajikan pendekatan baru untuk masalah kerunutan dengan menciptakan representasi formal ontological dari *software documents* dengan *source code*. Representasi yang dihasilkan kemudian disesuaikan untuk membangun link kerunutan (*Kerunutan link*) pada tingkat semantik.

*TLs,* pada penelitian yang dilakukan *Juergen Rilling dkk,* menghubungkan antara *software documents* yang berisi mulai dari *Software Requierement Specification* (SRS), *Software Design Documentation* (SDD) dihubungkan langsung dengan *Source Code,* sedangkan menurut *Héctor García dkk* terlihat di gambar 2.3, TLs tidak terhubung langsung pada SRS tetapi melalui Diagram Kelas. Sedangkan menurut *Mark Grechanik dkk* TLs menghubungkan dari Diagram Kasus Pengguna dengan *Source Code*.

Pada rencana penelitian yang ditawarkan, akan menghubungkan Cerita pengguna *(US)* dengan *Java* Kode sumber *(SC)* dengan mencari kemiripan kata yang ada pada US dan SC menggunakan *String Matching* yang telah tersedia dalam *PostgreSQL*, *PostgreSQL* dalam kakas terpisah menyediakan fungsi similaritas untuk mencocokkan dua kata yang berbeda dengan algoritma *trigram*.

Tabel 2.1 Perbandingan penelitian sebelumnya

| Penulis | SC TLs | Metode |
|---|---|---|
| Juergen Rilling | Dok. PL dan Kode Sumber | Ontologi |
| Mark Grechanik | Diagram Kasus Pengguna dan Kode Sumber | Ontologi |
| Héctor García | Diagram Kelas dan Kode Sumber | Kecocokan |

## 2.6. Sistem Temu Kembali Informasi

Sistem Temu Kembali Informasi atau *Information Retrieval* (IR) merupakan sistem yang berfungsi untuk menemukan informasi yang relevan dengan kebutuhan pemakai. Salah satu hal yang perlu diingat adalah bahwa informasi yang diproses terkandung dalam sebuah dokumen yang bersifat tekstual. Dalam konteks ini, temu kembali informasi berkaitan dengan representasi, penyimpanan, dan akses terhadap dokumen representasi dokumen. Dokumen yang ditemukan tidak dapat dipastikan apakah relevan dengan kebutuhan informasi pengguna yang dinyatakan dalam *query*. Pengguna Sistem Temu Kembali informasi sangat bervariasi dengan kebutuhan informasi yang berbeda-beda. Tahapan yang akan digunakan dalam IR secara umum adalah sebagai berikut:

- Indexing
- Searching
- Perengkingan relevansi keyword query

Konsep dasar dalam Information Retrieval System terdiri dari Indexing, Searching dan perengkingan relevansi keyword query. Dimana proses indexing dilakukan untuk membentuk database index terhadap koleksi dokumen yang dimasukkan, atau dengan kata lain, indexing merupakan proses persiapan yang dilakukan terhadap dokumen sehingga dokumen siap untuk retrive. Proses indexing sendiri meliputi 2 proses, yaitu dokumen indexing dan term indexing. Dari term indexing akan dihasilkan koleksi kata yang akan digunakan untuk meningkatkan performansi pencarian pada tahap selanjutnya. Tahap-tahap dalam proses indexing ialah:

- Word Token / Parsing
- Stopword Removal / filtering
- Stemming
- TF/IDF ( Term Frequency – Inversed Document Frequency )

### *2.6.1. Word Token/Parsing*

Untuk pemrosesan, dokumen dipilih menjadi unit-unit yang lebih kecil contohnya berupa kata, frasa atau kalimat. Unit hasil pemrosesan disebut sebagai token. Dalam proses ini biasanya juga digunakan sebuah daftar kata yang tidak digunakan (stoplist) karena tidak signifikan dalam membedakan dokumen atau kueri, misalnya kata-kata tugas seperti yang, hingga, dan dengan. Proses parsing akan menghasilkan daftar istilah beserta informasi tambahan seperti frekuensi dan posisi yang akan digunakan dalam proses selanjutnya (Mark , 2007).

### *2.6.2. Parsing*

Dalam bidang linguistik dan tata bahasa, parsing merupakan suatu proses yang dilakukan untuk memilah dokumen menjadi unit-unit yang lebih kecil berupa kata atau frasa . Unit hasil pemrosesan disebut sebagai token. Dalam proses ini biasanya juga digunakan sebuah daftar kata yang tidak digunakan (*stoplist*) karena tidak signifikan dalam membedakan dokumen atau kueri, misalnya kata-kata tugas seperti *yang*, *hingga*, dan *dengan*. Proses parsing akan menghasilkan daftar istilah beserta informasi tambahan seperti frekuensi dan posisi yang akan digunakan dalam proses selanjutnya.

### *2.6.3. Stemming*

*Stemming* adalah proses penghilangan prefiks dan sufiks dari kueri dan istilah-istilah dokumen. *Stemming* dilakukan atas dasar asumsi bahwa kata-kata yang sama memiliki makna yang serupa. Dalam hal keefektifan stemming dapat meningkatkan *recall* dengan mengurangi bentuk-bentuk kata ke bentuk kata dasarnya. Selain itu proses stemming juga dapat mengurangi ruang penyimpanan indeks (Mark , 2007).

Stemming adalah salah satu cara yang digunakan untuk meningkatkan kemampuan IR dengan cara mentransformasi kata-kata dalam kalimat atau dokumen teks menjadi kata dasar atau kata akarnya. Algoritma Stemming yang banyak digunakan ditujuka untuk bahasa Inggris namun bahasa Inggris memiliki morfologi yang berbeda dengan Bahasa Indonesia sehingga algoritma stemming untuk kedua bahasa tersebut juga berbeda. Proses stemming pada teks berbahasa Indonesia lebih rumit/kompleks karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan kata dasar dari sebuah kata. Beberapa algoritma stemming Bahasa Indonesia telah dikembangkan sebelumnya. Penggunaan algoritma stemming yang sesuai mempengaruhi performa sistem IR. Dalam penelitian *Jelita* telah membandingkan lima algoritma stemming dengan hasil perbandingan: Nazief yang terbaik dengan tingkat kebenaran 93% dari seluruh kata yang diuji dan 92% dari kata-kata yang unik, pada peringkat berikutnya adalah *Ahmad* dengan nilai 88.8%, *Idris* 87.9%, *Arifin* 87.7%, dan *Vega* 66.3%. *Vega* memiliki nilai terendah karena satu-satunya yang tidak menggunakan kamus. (*jelita, 2005)*

Algoritma-algoritma stemming memiliki kelebihan dan kekurangannya masing-masing. Efektifitas algoritma stemming dapat diukur berdasarkan beberapa parameter, seperti kecepatan proses, keakuratan, dan kesalahan. Menurut Dalam tulisan ini, penulis akan membandingkan efektifitas algoritma *Nazief* dan *Adriani* dengan algoritma *Porter* untuk proses stemming pada teks berBahasa Indonesia. Kesimpulan dari peneletian menyebutkan bahwa:

- Proses stemming dokumen teks berBahasa Indonesia menggunakan Algoritma Porter membutuhkan waktu yang lebih singkat dibandingkan dengan stemming menggunakan Algoritma Nazief & Adriani.
- Proses stemming dokumen teks berBahasa Indonesia menggunakan Algoritma Porter memiliki prosentase keakuratan (presisi) lebih kecil dibandingkan dengan stemming menggunakan Algoritma Nazief & Adriani.
- Pada proses stemming menggunakan Algoritma Nazief & Adriani, kamus yang digunakan sangat mempengaruhi hasil stemming. Semakin lengkap kamus yang digunakan maka semakin akurat pula hasil stemming.
- Kamus yang digunakan mempengaruhi perhitungan presisi. Semakin lengkap kamus yang digunakan maka semakin akurat pula nilai presisinya.

Dari kesimpulan tersebut maka penulis akan menggunakan *Algoritma Nazief & Adriani* karena penulis membutuhkan akurasi daripada kecepatan. Selain itu penulis mencoba untuk menggunakan kamus resmi bahasa Indonesia dalam *Kamus Besar Bahasa Indonesia* (KBBI) untuk menghasilkan kata dasar yang lebih akurat.

Algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut:

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tesebut adalah kata dasar., maka algoritma berhenti.
2. Inflection Suffixes (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa particles (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus Possesive Pronouns (“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus Derivation Suffixes (“-i”, “-an” atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a.
   a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika

kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.

b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.

4. Hapus Derivation Prefix. Jika pada langkah 3 ada sufiks yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
   a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
   b. For i = 1 to 3, tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Melakukan Recoding.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai kata dasar.
7. Proses selesai.

Tipe awalan ditentukan melalui langkah-langkah berikut:

1. Jika awalannya adalah: “di-”, “ke-”, atau “se-” maka tipe awalannya secara berturut-turut adalah “di-”, “ke-”, atau “se-”.
2. Jika awalannya adalah “te-”, “me-”, “be-”, atau “pe-” maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.
3. Jika dua karakter pertama bukan “di-”, “ke-”, “se-”, “te-”, “be-”, “me-”, atau “pe-” maka berhenti.
4. Jika tipe awalan adalah “none” maka berhenti. Jika tipe awalan adalah bukan “none” maka awalan dapat dilihat pada Tabel. 2.3

Tabel 2.2. Hapus awalan jika ditemukan.

| Awalan | Akhiran yang tidak diijinkan |
|---|---|
| be- | -i |
| di- | -an |
| ke- | -i, - kan |
| me- | -an |
| se- | -i, -kan |

Tabel 2.3. Cara Menentukan Tipe Awalan Untuk awalan “te-”

| Set 1 | Set 2 | Set 3 | Set 4 | Tipe Awalan |
|---|---|---|---|---|
| “-r-“ | “-r-“ | – | – | none |
| “-r-“ | | – | – | ter-luluh |
| “-r-“ | not (vowel or “-r-”) | “-er-“ | vowel | ter |
| “-r-“ | not (vowel or “-r-”) | “-er-“ | not vowel | ter- |
| “-r-“ | not (vowel or “-r-”) | not “-er-“ | – | ter |
| not (vowel or “-r-”) | “-er-“ | vowel | – | none |
| not (vowel or “-r-”) | “-er-“ | not vowel | – | te |

Tabel 2.4. Cara Menentukan Tipe Awalan Untuk awalan “te-”

| Tipe Awalan | Awalan yang harus dihapus |
|---|---|
| di- | di- |
| ke- | ke- |
| se- | se- |
| te- | te- |
| ter- | ter- |
| ter-luluh | ter |

***Hubungan Kerunutan***

Kerunutan adalah properti dari desain dan pengembangan perangkat lunak yang menghubungkan setiap artefak abstrak dengan artefak teknis maupun sebaliknya (Mens, 2008). Pada proyek perangkat lunak skala besar, kerunutan sangat penting untuk mengetahui hubungan kerunutan antar artefak dalam fase-fase yang berbeda (analisis kebutuhan, desain, dan implementasi). Selain itu, dapat mengetahui hubungan kerunutan antara artefak dan pihak pengembang yang terlibat.

Proses pembangunan ketelusuran yang dilakukan secara manual akan menghabiskan banyak waktu dan sangat berpotensi terjadi kesalahan sehingga informasi menjadi tidak lengkap. Oleh karena itu, sebuah sistem Kerunutan secara otomatis diperlukan untuk membangun ketelusuran antar artefak Penelitian tentang pendekatan untuk membangan Kerunutan telah dimulai sejak tahun 1984 oleh

*Dorfman dan Flynn* dengan membangun kakas untuk kerunutan dengan nama *ART (Automated Requirements Traceabillity)* (Dorfman, 1984). Hingga saat ini penelitian Kerunutan masih terbuka dan masih memiliki banyak tantangan, terutama dalam meminimalisasi kesalahan keterhubungan antarartefak (Kannenberg, 2009). Penelitian yang dilakukan oleh *Siti Rochimah, dkk* pada tahun 2007 mengevaluasi berbagai pendekatan untuk kerunutan dengan menggunakan kerangka kerja taksonomi evolusi perangkat lunak yang diusulkan oleh Buckley (Rochimah, 2007).

Hubungan kerunutan dengan artefak secara tekstual diperlukan untuk menghubungkan cerita pengguna dengan kode sumber menggunakan pencocokan string dengan kemiripan.

## 2.7. Similarity

Similarity atau similaritas merupakan tingkat perbandingan persentase kemiripan antar dokumen yang diuji. Similarity ini akan menghasilkan nilai dimana nilai tersebut yang akan dijadikan acuan dalam menentukan persentase tingkat kemiripan sebuah dokumen yang diuji. Besarnya persentase similarity akan dipengaruhi oleh tingkat kemiripan dari dokumen yang diuji semakin besar persentase *similarity* maka semakin tinggi tingkat kemiripan dokumen

K-Gram adalah rangkaian *terms* dengan panjang K. Kebanyakan yang digunakan sebagai *terms* adalah kata. K-Gram merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode K-Gram ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen. Dalam Markov Model nilai K-Gram yang sering digunakan yaitu, 2-gram (bigram), 3-gram (trigram) dan seterusnya disebut K-Gram (4-gram, 5-gram dan seterusnya). Dalam *natural language processing*, penggunaan K-Gram (atau lebih dikenal dengan n-gram), proses *parsing token* (tokenisasi) lebih sering menggunakan 3-gram dan 4-gram, sedangkan 2-gram digunakan dalam parsing sentence, misal dalam part-of-speech (POS). Penggunaan 2-gram dalam tokenisasi akan menyebabkan tingkat perbandingan antar karakter akan semakin besar. Contohnya pada kata „makan‟ dan „mana‟ yang merupakan dua kata yang sama sekali berbeda. Dengan menggunakan metode bigram dalam mencari similarity, hasil dari bigram tersebut yaitu kata

“makan” akan menghasilkan bigram ma, ak, ka, an serta kata “mana” akan menghasilkan bigram ma, an, na. Dengan demikian, akan terdapat kesamaan dalam pengecekannya similarity. Namun jika menggunakan 3-gram (“makan” = mak, aka, kan dan “mana” = man, ana) atau 4-gram (“makan” = maka, akan, dan “mana” = mana) akan mengecilkan kemungkinan terjadinya kesamaan pada kata yang strukturnya berbeda.

## 2.8. Konvensi Penamaan pada Java

Konvensi penamaan pada *java* dibuat agar kode program lebih mudah dimengerti dan lebih mudah dibaca. Konvensi ini juga dapat memberikan informasi tentang fungsi dari identifier, apakah itu konstanta, paket, atau kelas. Aturan konvensi penamaan telah ditetapkan oleh *Sun Microsystems, Inc.* seperti pada tabel 2.2. Konvensi Penamaan pada *java.* (Sun, 1995-1999)

Tabel 2.5. Konvensi penamaan pada java

| Tipe *Indentifier* | Aturan Penamaan | Contoh |
|---|---|---|
| *Packages* | Awalan (*prefix)* nama paket yang unik selalu ditulis dalam huruf ASCII semua huruf kecil dan harus menjadi salah satu nama top-level domain, saat ini *com, edu, gov, mil, net, org*, atau salah satu dari kode dua huruf mengidentifikasi negara-negara sebagaimana ditentukan dalam *ISO Standard 3166, 1981*.<br><br>komponen berikutnya dari nama paket bervariasi sesuai dengan konvensi penamaan internal organisasi sendiri. konvensi tersebut mungkin ditentukan bahwa komponen nama direktori tertentu menjadi divisi, departemen, proyek, mesin, atau nama login. | com.sun.eng<br>com.apple.quicktime.v2<br>edu.its.tesis |
| *Classes* | Nama kelas diberi nama kata benda, apabila lebih dari satu kata dengan huruf pertama dari setiap kata menggunakan huruf kapital. gunakan nama kelas dengan nama yang sederhana dan deskriptif. Gunakan seluruh kata-hindari | class Raster;<br>class ImageSprite;<br>class UserStory; |

| | | |
|---|---|---|
| | akronim dan singkatan (kecuali singkatan yang jauh lebih banyak digunakan daripada bentuk panjang, seperti URL atau HTML). | |
| *Interfaces* | Nama *interface* sama dengan nama kelas dengan huruf capital pada huruf pertama setiap kata | interface RasterDelegate;<br>interface Storing; |
| *Methods* | Metode harus diberi nama dengan kata kerja, apabila lebih dari satu kata gunakan huruf kecil pada kata pertama, gunakan huruf capital pada setiap huruh pertama dari setiap kata berikutnya | run();<br>runFast();<br>getBackground();<br>simpanCeritaPengguna(); |
| *Variables* | Nama variabel, semua *instance*, kelas, dan konstanta kelas apabila lebih dari satu kata gunakan huruf kecil pada kata pertama, gunakan huruf capital pada setiap huruh pertama dari setiap kata berikutnya, nama variabel tidak harus dimulai dengan garis bawah _ atau tanda dolar $ karakter, meskipun keduanya diizinkan.<br><br>Nama variabel harus pendek namun bermakna. Pemilihan nama variabel harus mnemonic- yaitu, dirancang agar pembaca mengerti maksud penggunaannya. Nama variabel satu karakter harus dihindari kecuali untuk sementara ("sekali pakai") variabel. Nama-nama umum variabel sementara contoh *i, j, k, m,* dan *n* untuk bilangan bulat; *c, d,* dan *e* untuk karakter. | int i;<br>char c;<br>float myWidth;<br>String kataDasar; |
| *Constants* | Nama-nama konstanta kelas variabel yang dideklarasikan dan konstanta ANSI harus semua huruf kapital dengan kata-kata yang dipisahkan oleh garis bawah ("_"). (Konstanta ANSI harus dihindari, untuk kemudahan debugging.) | static final int MIN_WIDTH = 4;<br>static final int MAX_WIDTH = 999;<br>static final int GET_THE_CPU = 1; |

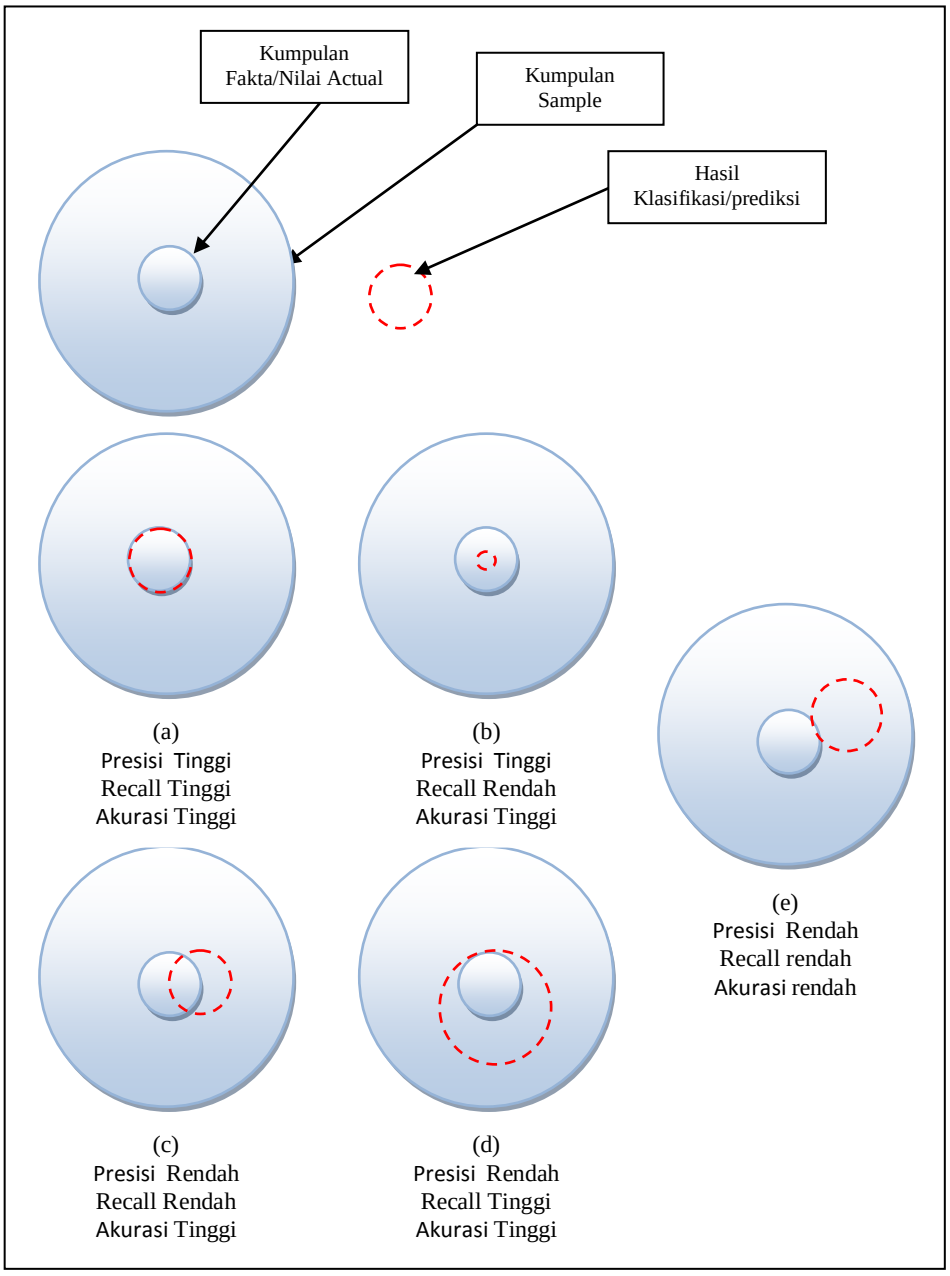
## 2.9.Metode Evaluasi Sistem

Dalam pengenalan pola (*pattern recognition*) dan temu kembali informasi (*information retrieval*) dengan klasifikasi biner, *precision/*presisi (juga disebut nilai

prediktif positif) adalah fraksi contoh diambil yang relevan atau tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh system. Sementara *Recall* (juga dikenal sebagai sensitivitas) adalah fraksi contoh yang relevan yang diambil atau tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Sedangkan akurasi didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual.

Kedua presisi dan recall karena itu didasarkan pada pemahaman dan ukuran relevansi. Dalam pemodelan prediksi, akurasi dan presisi memiliki arti yang berbeda. Untuk presisi, recall.dan akurasi dapat kita lihat pada gambar 2.6. Pada gambar tersebut terdapat tiga bagian yaitu lingkaran paling besar menunjukkan kumpulan sample atau data, sedangkan lingkaran kecil yang ditengan menunjukkan kumpulan fakta atau nilai actual atau fraksi yang sesungguhnya, sedangkan lingkaran putus-putus mewakili hasil klasifikasi atau prediksi atau keluaran dari sebuah system atau perhitungan otomatis.

Pada gambar 2.6 bagian (a) menunjukan bahwa lingkaran putus-putus tepat berada pada lingkaran kecil yang berada di tengah hal ini menunjukkan hasil dari prediksi atau perhitungan otomatis mendekati sempurna dibandingkan dengan kumpulan nilai-nilai actual sehingga disimpulkan nilai presisi tinggi, recall tinggi, dan akurasi tinggi. Gambar bagian (b) menunjukan bahwa lingkaran putus-putus tepat berada pada lingkaran kecil yang berada di tengah tetapi tidak secara keseluruhan lingkaran terlingkupi hal ini menunjukkan hasil dari prediksi atau perhitungan otomatis tidak ada yang salah akan tetapi tidak semua nilai actual dapat terprediksi, sehingga disimpulkan nilai nilai presisi tinggi, recall rendah, dan akurasi tinggi.




Gambar 2.6 Tingkat Presisi dan Akurasi

Gambar bagian *(c)* menunjukan bahwa lingkaran putus-putus tidak tepat berada pada lingkaran kecil yang berada di tengah dan tidak secara keseluruhan lingkaran terlingkupi hal ini menunjukkan hasil dari prediksi atau perhitungan otomatis masih terdapat kesalahan, dan tidak semua nilai actual dapat terprediksi, sehingga disimpulkan nilai nilai presisi rendah, recall rendah, dan akurasi tinggi. Gambar bagian *(d)* menunjukan bahwa lingkaran putus-putus melingkari semua lingkaran kecil yang berada di tengah tetapi melebihi lingkaran kecil tersebut, hal ini

menunjukkan hasil dari prediksi atau perhitungan otomatis dapat mendeteksi semua nilai actual tetapi masih terdapat kesalahan nilai sample yg tidak diharapkan juga ikut terdeteksi, sehingga disimpulkan nilai presisi rendah, recall tinggi, dan akurasi tinggi. Gambar bagian *(e)* menunjukan bahwa lingkaran putus-putus tidak melingkari semua lingkaran kecil yang berada di tengah, hal ini menunjukkan hasil dari prediksi atau perhitungan otomatis tidak dapat mendeteksi semua nilai actual tetapi mendeteksi nilai yang tidak dikehendaki, sehingga disimpulkan nilai presisi rendah, recall rendah, dan akurasi rendah.

Untuk memudahkan pemahaman pada presisi, recall dan akurasi dapat dilihat pada tabel 2.3. tentang rumusan presisi, recall dan akurasi serta dapat dilihat pada ilustrasi contoh misalnya sebuah program komputer untuk mengenali truk yang lewat dijalanan melalui sebuah CCTV, dalam video terdapat 100 truk dan 900 kendaraan lain (bukan truk). Hasilnya program komputer tersebut medeteksi 110 kendaraan sebagai truk, 110 kendaraan yg dideteksi sebagai truk tersebut dicek kembali oleh manusia, ternyata dari 110 kendaraan tersebut hanya 90 yang merupakan truk, sedangkan 20 lainnya merupakan kendaraan lain.

Dari kasus tersebut maka dapat disimpulkan bahwa program komputer tersebut memiliki precision sebesar 82%, recall sebesar 90% dan accuracy sebesar 97% yang didapatkan dari perhitungan berikut:

Tabel 2.6. Rumusan precision recall dan accuracy

| | | Nilai Sebenarnya | | Total |
|---|---|---|---|---|
| | | True | False | |
| Nilai Prediksi | True | TP (true positive) relevan/hasil yg benar | FP (false positive) tidak relevan/hasil tidak diharapkan | TP + FP |
| | False | FN (false negative) relevan, hasil yg hilang | TN (true negative) tidak relevan, hasil yg tdk benar | FN+TN |
| | Total | TP+FN | FP+TN | TP+FP+FN+TN |

$$Precision = \frac{TP}{TP + FP} \quad (2.1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.2)$$

$$Accuracy = \frac{TP + TN}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \quad (2.3)$$

**Keterangan:**

- TP (True Positive)
- FP (False Positive)
- FN (False Negative)
- TN (True Negative)

Sehingga berdasarkan rumus diatas dengan kasus deteksi truk dapat dihitung sebagai berikut:

| | | Nilai Sebenarnya | | Total |
|---|---|---|---|---|
| | | True | False | |
| Nilai Prediksi | True | 90 | 20 | 110 |
| | False | 10 | 880 | 890 |
| | Total | 100 | 1000 | 1100 |

$$Precision = \frac{90}{90+20} = \frac{90}{110} = 0.82 = 82\%$$

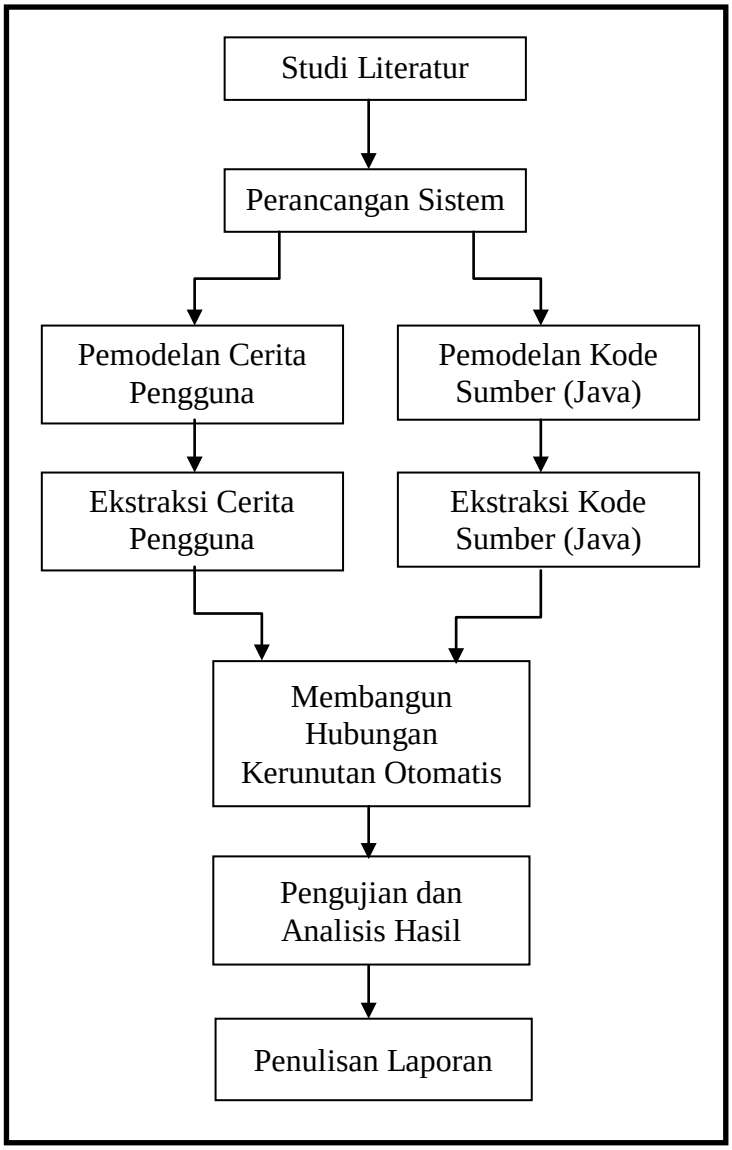
$$Recall = \frac{90}{90+10} = \frac{90}{100} = 0.9 = 90\%$$

$$Accuracy = \frac{90+880}{90+20+10+880} = \frac{970}{1100} = 0.97 = 97\%$$

# BAB 3
# METODOLOGI PENELITIAN

## 3.1. Metodologi Penelitian

Secara umum, penelitian ini diawali dengan studi literatur, analisis dan desain, implementasi, serta diakhiri dengan pengujian. Sedangkan penulisan laporan tesis dimulai pada awal sampai akhir penelitian. Secara lebih detail, penelitian ini dirancang dengan urutan pada gambar 3.1 sebagai berikut.




Gambar 3.1 Metodologi Penelitian

Pada gambar 3.1 metodologi penelitian digambarkan sistematika metodologi penelitian yang dilakukan adalah sebagai berikut.

1. Studi Literatur

   Studi literatur ini dilakukan untuk mendapatkan informasi dari literatur-literatur yang berhubungan dengan penelitian ini, serta perkembangan metode yang telah dilakukan dalam melakukan penelusuran kerunutan hubungan antar artefak perangkat lunak sampai dengan kode sumber. Penelitian sebelumnya (Juergen Rilling, 2007) membangun otomatisasi hubungan kerunutan antara dokumen perangkat lunak dengan kode sumber menggunakan ontologi, (Mark Grechanik, 2007) membangun otomatisasi hubungan kerunutan antara diagram kasus pengguna dengan kode sumber menggunakan ontologi, (Héctor García, 2008) membangun otomatisasi hubungan kerunutan antara diagram kelas dengan kode sumber menggunakan kococokan.

2. Perancangan Sistem

   Perancangan sistem dilakukan untuk merancang sistem yang dapat melakukan otomatisasi kerunutan hubungan antara cerita pengguna denga kode sumber .

3. Pemodelan Cerita Pengguna

   Pemodelan Cerita Pengguna digunakan untuk membuat model basisdata dari Cerita Pengguna yang telah dibuat dalam format kalimat bebas, pemodelan disini akan memberikan kode setiap Cerita Pengguna dari setiap proyek, serta membuat model database untuk menyimpan setiap kata penting dalam Cerita Pengguna.

4. Ekstraksi Cerita Pengguna

   Ekstraksi cerita pengguna adalah proses merubah kalimat bebas dalam Cerita Pengguna menjadi daftar kata dasar dalam table basisdata menggunakan Algoritma *Nazief dan Adriani*.

5. Pemodelan Kode Sumber

   Pemodelan Kode Sumber digunakan untuk membuat model basisdata dari kode sumber java yang telah dibuat oleh *programmer*, pemodelan disini akan memberikan kode setiap *file* kode sumber dari setiap proyek, serta membuat model basisdata untuk menyimpan setiap kata dalam kode sumber.

6. Ekstraksi Kode Sumber

Ekstraksi Kode Sumber adalah proses parsing dari kode sumber menjadi daftar nama class, method dan attribute dari setiap class.

7. Membangun Hubungan Kerunutan Otomatis

   Membangun Kerunutan Hubungan antara tabel Cerita Pengguna dan tabel kode sumber menggunakan *string matching* dengan algoritma *trigram* yang telah diintegrasikan dengan postgreSQL (*pg_trgrm*). Indentifikasi adanya hubungan antara US dan SC menggunakan query ini akan didapat kesimpulan apakah US dan SC telah terhubung atau belum dalam bentuk matrik kerunutan hubungan.

8. Pengujian dan Analisis Hasil

   Hal ini dilakukan untuk menguji apakah sistem dapat dijadikan acuan untuk menghasilkan matrik kerunutan hubungan yang benar dan sesuai dengan proses pembuatan matrik kerunutan hubungan secara manual.

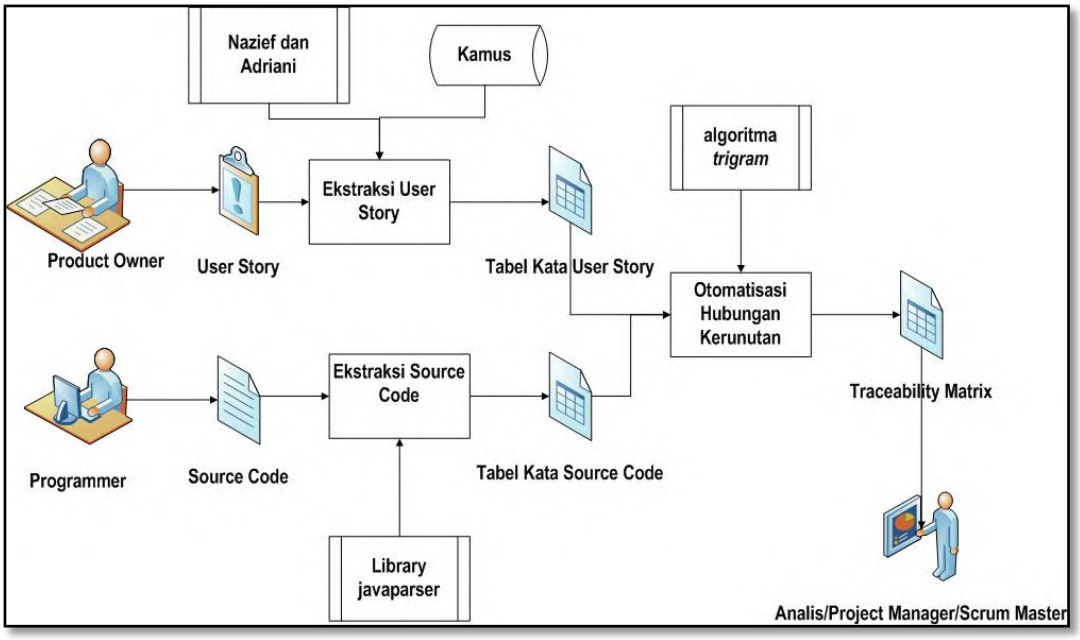
9. Penyusunan Laporan

   Hal ini dilakukan untuk melakukan penggambaran dari hasil penelitan yang telah dilakukan.

## 3.2. Perancangan Sistem

Pada bagian ini menjelaskan langkah-langkah dari sistem yang akan dibangun. Pada Gambar 3.2 dijelaskan alur dari sistem dalam membangun hubungan kerunutan yang menghasilkan matrix hubungan kerunutan. Sistem memiliki dua masukan yaitu cerita pengguna dari pengguna dan kode sumber dari programmer, sedangkan luaran yang dihasilkan adalah matrik kerunutan.

Mekanisme untuk menghasilkan matrik kerunutan hubungan diperlukan tiga proses utama yaitu proses pertama adalah ekstraksi kata dari cerita pengguna yang ditulis secara bebas berdasarkan format penulisan cerita pengguna. Proses kedua adalah ekstraksi kata dari kode sumber yang telah dibuat oleh *programmer*. Proses ketiga adalah mencocokan kata yang ada pada table cerita pengguna dengan daftar kata yang ada pada kode sumber apabila terdapat kesamaan kata pada tingkat tertentu maka ditentukan terjadi hubungan kerunutan antara cerita pengguna dengan kode sumber tersebut, selanjutnya cerita pengguna dengan kode sumber yang terhubung tersebut ditandai dalam matrik hubungan kerunutan. Gambaran umum dari perancangan sistem digambarkan pada Gambar 3.2.




Gambar 3.2 Perancangan Sistem

Dari gambar 3.2 dapat dijelaskan peran dari masing-masing blok diagram diatas sebagai berikut:

- Product Owner

  Product Owner bertanggung-jawab untuk memaksimalkan nilai produk dan hasil kerja Tim Pengembang. Cara pelaksanaannya sangat bervariasi antar organisasi, Tim Scrum dan individu. Product Owner merupakan satu-satunya orang yang bertanggung-jawab untuk mengelola Product Backlog. Pengelolaan Product Backlog mencakup:

  - Mengekspresikan dengan jelas item Product Backlog;
  - Mengurutkan item di dalam Product Backlog untuk mencapai tujuan dan misi dengan cara terbaik;
  - Mengoptimalkan nilai dari hasil pekerjaan Tim Pengembang;
  - Memastikan Product Backlog transparan, jelas, dan dapat dilihat semua pihak, dan menunjukkan apa yang akan dikerjakan oleh Tim Scrum selanjutnya;
  - Memastikan Tim Pengembang dapat memahami item dalam *Product Backlog* hingga batasan yang diperlukan;

*Product Owner* dapat saja mengerjakan pekerjaan-pekerjaan di atas, atau menyerahkan pengerjaannya kepada Tim Pengembang, namun satu-satunya pihak yang bertanggung jawab tetaplah Product Owner. Product Owner adalah satu orang dan bukan berupa sebuah komite. Product Owner dapat mengejawantahkan aspirasi dari komite ke dalam Product Backlog, namun mereka yang ingin erubah prioritas item Product Backlog, harus melakukannya melalui Product Owner.

*Product owner* bersama *stakeholder* yang lain menyampaikan kebutuhan system kepada tim yang selanjutnya tim *scrum* membantu menuliskan kebutuhan system tersebut dalam bentuk cerita pengguna yang delengkapi dengan kriteria penerimaan (*acceptance criteria*).

Programmer/Tim Pengembang

*Programmer* atau didalam *scrum* disebut tim pengembang (*Development Team*) yang terdiri dari para profesional yang bekerja untuk menghasilkan tambahan potongan produk (selanjutnya disebut Inkremen) “Selesai”, yang berpotensi untuk dirilis di setiap akhir Sprint. Hanya anggota Tim Pengembang yang mengembangkan Inkremen ini.

Tim Pengembang dibentuk dan didukung oleh organisasi untuk mengatur dan mengelola pekerjaannya secara mandiri. Sinergi yang ada di dalam tim akan meningkatkan efisiensi dan efektifitas dari Tim Pengembang secara keseluruhan.

Tim Pengembang memiliki karakteristik sebagai berikut:

- Mereka mengatur dirinya sendiri. Tidak ada satu orang pun (bahkan *Scrum Master*) yang memerintah Tim Pengembang bagaimana cara merubah Product Backlog menjadi Inkremen yang berpotensi untuk dirilis;
- Tim Pengembang berfungsi antar-lintas, sebagai sebuah tim, memiliki semua keahlian yang dibutuhkan untuk menghasilkan produk;
- Scrum tidak mengenal adanya jabatan tertentu untuk anggota Tim Pengembang selain Pengembang, apapun pekerjaan yang dikerjakan oleh masing-masing anggota tim; tidak ada pengecualian untuk aturan yang satu ini;

- Tim Pengembang tidak mengenal adanya sub-tim yang dikhususkan untuk bidang tertentu seperti pengujian atau analisa bisnis; tidak ada pengecualian untuk aturan yang satu ini;
- Anggota Tim Pengembang boleh memiliki spesialisasi keahlian dan fokus di satu area tertentu, namun akuntabilitas dari hasil dari pekerjaan secara keseluruhan adalah milik Tim Pengembang.

Dalam kasus ini programmer atau tim pengembang membuat kode program sesuai dengan cerita pengguna yang telah dibuat sebelumnya.

Cerita Pengguna

Pada tahun 1999, Kent Beck mulai memperkenalkan istilah cerita pengguna untuk fitur produk perangkat lunak. Dia menggambarkan bahwa cerita pengguna dituliskan dari perspektif pengguna mengenai apa yang pengguna inginkan pada system serta apa kelebihan yang sistem bisa lakukan untuknya. Dengan demikian, pandangan benar-benar berubah dari produk ke pengguna dan Cerita Pengguna menjadi de facto standar untuk kebutuhan pada *Agile framework*.

Dalam proyek-proyek Scrum, Product Backlog adalah daftar cerita pengguna. Cerita Pengguna ini diberi skala prioritas dan dibawa ke *Backlog Sprint* dalam meeting *Sprint Planning*. Perkiraan waktu juga didasarkan pada cerita pengguna dan ukuran besaran produk juga diperkirakan melalui Cerita pengguna .

Cerita Pengguna merupakan artefak utama dalam pengembangan perangkat lunak pada lingkungan pengembangan cepat (*agile*) terutama *scrum* dan *eXtreme Programming*. Cerita Pengguna merupakan definisi tertinggi dari kebutuhan perangkat lunak berisi informasi yang dibutuhkan untuk memperkirakan upaya dan waktu implementasinya.

Pada bagian ini cerita pengguna menjadi obyek utama penelitian yang akan diekstraksi menjadi potongan-potongan kata dasar yang nantinya akan dihubungkan dengan kode sumber, cerita pengguna dinyatakan dalam bentuk sebagai berikut seperti pada bagian 2.3:

As a <role> I can <activity> so that < value>

*Dimana*

- *Role* – menggambarkan *siapa* yang melakukan aksi
- *Activity* – Menggambarkan aksi yang dilakukan
- *Value* – Menggambarkan nilai yang dihasilkan dari aksi yang dilakukan

Sebagai Contoh:

"Sebagai Kasir Saya Dapat Mencetak Laporan Setiap Shift sehingga saya mengetahui berapa uang kas, kartu kredit maupun kartu debit yang saya terima".

Kode sumber

Seperti telah dijelaskan pada bagian 2.2 bahwa kode sumber adalah suatu rangkaian peryataan atau deklarasi yang ditulis dalam bahasa pemrograman komputer yang terbaca manusia. Kode sumber yang menyusun suatu program biasanya disimpan dalam satu atau lebih berkas teks.

Selain cerita pengguna, kode sumber juga merupakan bagian inti dari penelitian ini karena hasil ekstraksi kode sumber akan dihubungkan dengan hasil ekstraksi dari cerita pengguna yang hasilnya berupa matrik Kerunutan.

Ekstraksi Cerita Pengguna

Ekstraksi cerita pengguna adalah proses merubah kalimat bebas yang terdapat dalam cerita pengguna menjadi daftar kata dasar dalam table basis data. Proses ini dimulai dari mengkodekan masing-masing cerita pengguna, *parsing*, *stop word removal*, *stemming* menggunakan Algoritma *Nazief* dan *Andriani* hingga terbentuk kata dasar yang selanjutnya disimpan didalam table daftar kata cerita pengguna seperti yang akan dijelaskan pada bagian 3.3.

Ekstraksi Source Code

Ekstraksi Kode Sumber adalah proses parsing dari kode sumber menjadi daftar nama class, method dan attribute dari setiap class dalam kode sumber *java* menggunakan *library javaparser* yang akan dijelaskan pada bagian 3.4.

Tabel Kata cerita pengguna

Setiap kata hasil stemming akan disimpan dalam table daftar kata yang sudah berupa kata dasar, yang terdiri dari bagian *role, activity, value* dan *acceptance criteria* untuk masing-masing Cerita pengguna .

Tabel Kata Source Code

Hasil dari ekstraksi kode sumber berupa Nama Class, Nama Method dan Atribut akan disimpan kedalam Tabel Kode Sumber

Otomatisasi Hubungan kerunutan

Otomatisasi hubungan kerututan ini akan dibangun dengan cara menghubungkan daftar kata dalam cerita pengguna dengan kode sumber menggunakan *string matching* dengan algotirma *trigram*.

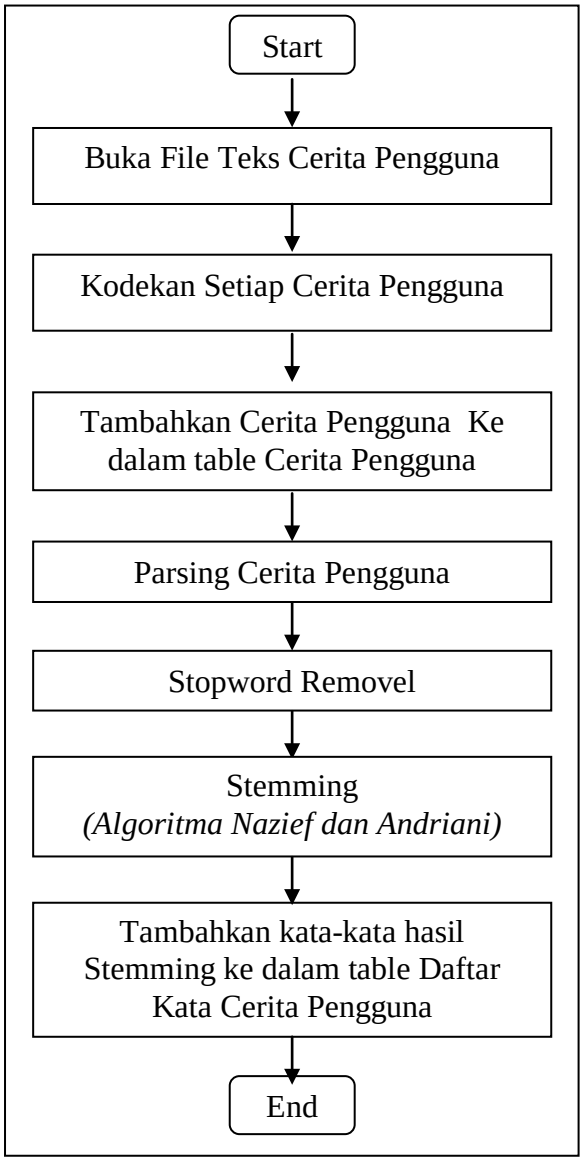
Matrix Hubungan Kerunutan

Hasil dari otomatisasi pembentukan hubungan kerunutan disajikan dalam bentuk matrik krunutan hubungan.

## 3.3.Ekstraksi Cerita Pengguna

Ekstraksi cerita pengguna adalah proses merubah kalimat bebas dalam cerita pengguna menjadi daftar kata dasar dalam table basis data seperti tampak pada gambar gambar 3.3, dengan langkah-langkah sebagai berikut:

1. Membuka File Teks cerita pengguna (US), file cerita pengguna disimpan dalam teks file dibuka dan selanjutnya dipilah per cerita pengguna .
2. Setiap US yang telah dipilah diberi kode unik sebagai identitas masing-masing US.
3. US yang telah diberi kode disimpan kedalam table database.
4. US di dalam table selanjutnya diparsing menjadi potongan-potongan kata dan membuang semua tanda baca.
5. Proses stopword removel membersihkan kata-kata hasil parsing dengan membandingkan setiap kata dengan daftar kata yang tidak digunakan (*stoplist*). Jika kata-kata ditemukan dalam stoplist maka kata-kata tersebut dihilangkan dari daftar kata yang akan digunakan.
6. Hasil dari proses 5 selanjutnya dilakukan stemming setiap pada kata dengan algoritma Nazief dan Andriani yang ditambahkan dengan daftar kata dalam kamus bahasa Indonesia (KBBI).
7. Setiap kata hasil stemming akan disimpan dalam table daftar kata yang sudah berupa kata dasar, yang terdiri dari bagian *role*, *activity*, *value* dan *acceptance criteria* untuk masing-masing Cerita pengguna .




Gambar 3.3 Ekstraksi cerita pengguna

3.4.Ekstraksi kode sumber

Ekstraksi Kode Sumber adalah proses parsing dari kode sumber menjadi daftar nama class, method dan attribute dari setiap class seperti tampak pada gambar gambar 3.4, dengan langkah-langkah sebagai berikut:

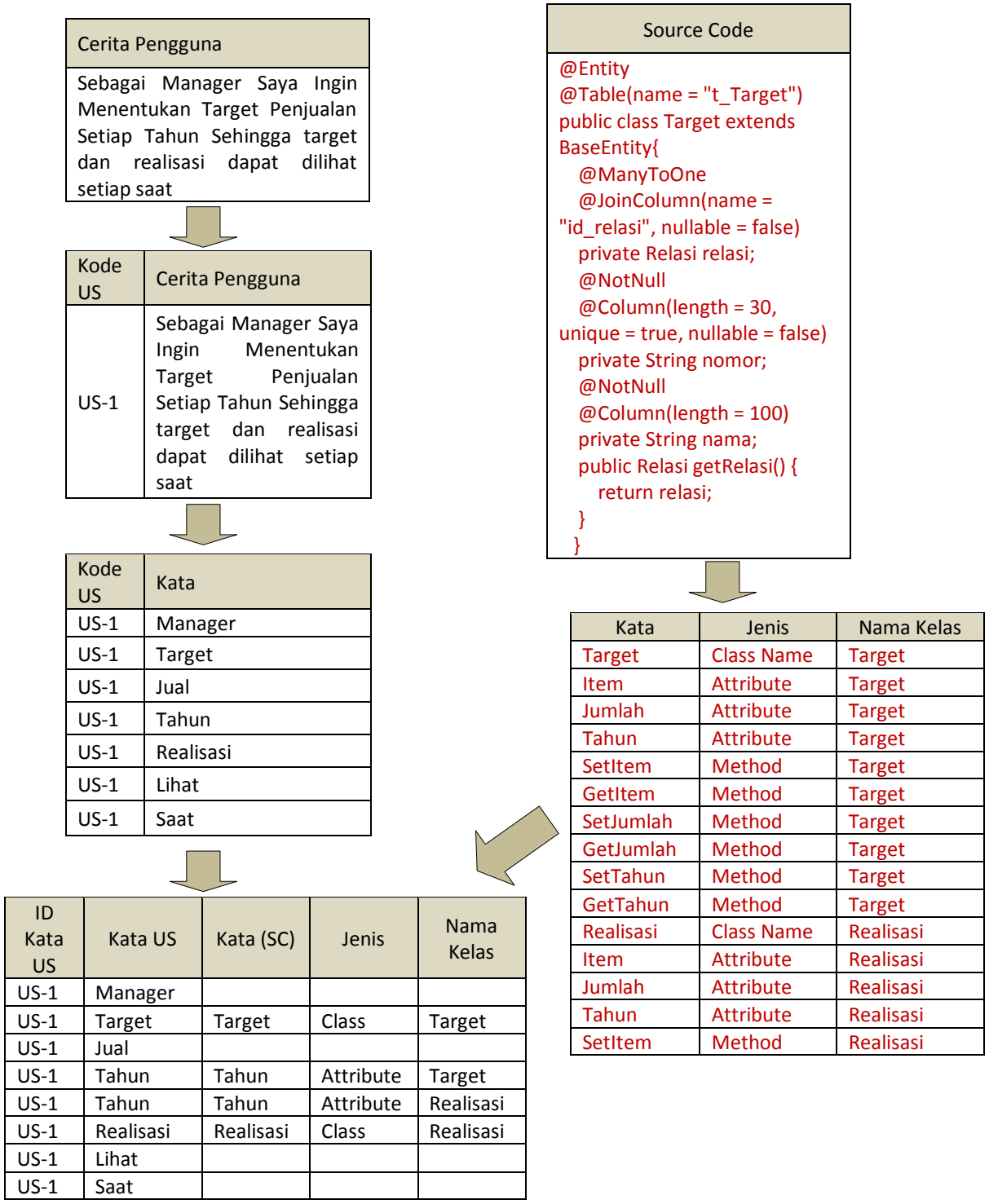
1. Membuka Folder Project untuk Kode Sumber (KS)
2. Buka setiap file KS didalam folder.
3. Parsing file KS menggunakan Parsing *library javaparser*

4. Simpan Hasil Parsing berupa Nama Class, Nama Method dan Atribut ke Dalam Tabel Kode Sumber


Gambar 3.4 Ekstraksi Kode Sumber

3.4.Membangun hubungan kerunutan

Membangun Kerunutan Hubungan antara tabel cerita pengguna dan tabel kode sumber menggunakan *string matching* dengan algoritma *trigram* yang telah diintegrasikan dengan postgreSQL (*pg_trgrm*). Indentifikasi adanya hubungan antara US dan SC menggunakan query ini akan didapat kesimpulan apakah US dan SC telah terhubung atau belum, seperti tampak pada gambar gambar 3.5, dengan langkah-langkah sebagai berikut:




Gambar 3.5 Ilustrasi Pembentukan Hubungan Kerunutan

*[Halaman ini sengaja dikosongkan]*

# BAB 4
# HASIL DAN PEMBAHASAN

## 4.1 Pengumpulan Dataset

Dataset yang digunakan adalah tiga project pembangunan perangkat lunak pada berbagai instansi pemerintah dan swasta

Dataset pertama diimplementasikan pada Dinas Pendapatan Pengelolaan Keuangan dan Aset Daerah pada salah satu kabupaten di Jawa Tengah. Aplikasi ini digunakan untuk memungut retribusi truk pengangkut galian C. Pada aplikasi ini digunakan IP camera sebagai alat untuk menangkap gambar truk yang selanjutnya diolah menggunakan HAAR training untuk mengetahui klasifikasi kendaraan yang memasuki pos penarikan retribusi. Aplikasi ini dibangun dengan bahasa pemrograman *java* desktop dengan *swing* yang selanjutnya disebut smartPortal.

Dataset yang kedua adalah aplikasi absensi online yang diimplementasikan pada salah satu Pemerintah Kota di Jawa Timur. Aplikasi ini digunakan untuk mengintegrasikan mesin-mesin absensi baik yang menggunakan *fingerprint* maupun *face recognition.* Data dari masing-masing mesin absensi setiap SKPD atau untit kerja secara otomatis terekam pada server database di Badan Kepegawaian Daerah yang selanjutnya disebut smartAbsensi.

Dataset yang ketiga adalah aplikasi tour dan travel yang diimplementasikan pada salah satu perusahaan tour travel di wilayah Jawa Timur. Aplikasi ini digunakan untuk melakukan resevasi dan pembelian tiket pesawat, paket tour, hotel, dan penyewaan mobil, yang selanjutnya disebut smartTravel.

*Identifeir* kode sumber pada semua dataset masih menggunakan bahasa campuran yaitu bahasa Inggris dan Indonesia, serta masih banyak terdapat singkatan-singkatan, sehingga hasil identifikasi system menjadi tidak optimal. Pada masa mendatang diharapkan terdapat dataset dengan penulisan kode sumber yang benar dan disertai cerita pengguna.

Tabel 4.1 Daftar Dataset

| No. | Nama Aplikasi | Ukuran | Jumlah Cerita Pengguna | Jumlah File Program Java |
|---|---|---|---|---|
| 1. | smartPortal | Kecil | 17 | 52 |
| 2. | smartAbsensi | Sedang | 47 | 161 |
| 3. | smartTravel | Besar | 80 | 222 |

## 4.2 Implementasi

Proses identifikasi kerunutan hubungan dilakukan secara otomatis. Proses otomatis dilakukan dengan menggunakan kakas bantu. Kakas bantu dibangun dengan bahasa pemrograman Java, sistem basis data PostgreSQL versi 9.4, proses ekstraksi kode sumber dibantu dengan *javaparser library* dan menerapkan teori yang sudah dipaparkan pada bab sebelumnya. Proses otomatis menyelesaikan tiga proses pada metodologi yang diusulkan dalam penelitian ini, yaitu proses ekstraksi cerita pengguna, proses ekstraksi kode sumber, dan proses otomatisasi hubungan kerunutan antara cerita pengguna dengan kode sumber.

### 4.2.1 Ekstraksi Cerita Pengguna

Proses ekstraksi cerita pengguna seperti yang telah dijelaskan pada bagian metodologi dimulai dari proses kodefikasi cerita pengguna, *parsing, stopword removel, stemming* menggunakan algoritma *Nazief dan Andrian* dan dibantu dengan KBBI (Kamus Besar Bahasa Indonesia), hasil dari ekstraksi ini disimpan kedalam table database *user_story* dengan struktur table seperti pada tabel 4.2.

Tabel 4.2 Tabel user_story

| Nama Kolom | Tipe data | Keterangan |
|---|---|---|
| id_project | character varying(10) | Kode proyek |
| id_user_story | character varying(10) | Kode cerita pengguna |
| user_story | Text | Isi cerita pengguna |

Sebagai contoh setelah system membaca seluruh file text cerita pengguna maka setiap serita pengguna disimpan pada tabel diatas, misalnya terdapat cerita pengguna sebagai berikut:

```
Sebagai pengguna saya ingin memiliki halaman login  sehingga
saya dapat masuk kedalam sistem menggunakan user dan password.
```

Gambar 4.1 Contoh cerita pengguna

Cerita pengguna pada gambar 4.1 akan disimpan kedalam tabel *user_story* seperti pada Tabel 4.3.

Tabel 4.3 Contoh isi tabel *user_story*

| id_project | id_user_story | user_story |
|---|---|---|
| P001 | P001-001 | Sebagai pengguna saya ingin memiliki halaman login sehingga saya dapat masuk kedalam sistem menggunakan user dan password. |

Proses selanjutnya adalah dari setiap baris tabel *user_story* akan dilakukan proses parsing menjadi potongan-potongan kata dan membuang semua tanda baca. Setelah proses parsing dilakukan proses membersihkan kata-kata hasil parsing yang tidak diperlukan dengan membandingkan setiap kata dengan daftar kata yang tidak digunakan (*stoplist*). Jika kata-kata ditemukan dalam stoplist maka kata-kata tersebut dihilangkan dari daftar kata yang akan digunakan. Kata-kata yang ditemukan selanjutnya dilakukan stemming setiap pada kata dengan algoritma *Nazief dan Andriani* yang ditambahkan dengan daftar kata dalam kamus besar bahasa Indonesia (KBBI). Proses ini dapat dilihat pada gambar 4.2 Potongan kode sumber untuk mengambil kata dasar dan proses menyimpan kedalam tabel daftar kata yang sudah berupa kata dasar kedalam tabel *user_story_kata* (Tabel 4.4).

Tabel 4. 4 Tabel *user_story_kata*

| Nama Kolom | Tipe data | Keterangan |
|---|---|---|
| id_user_story | character varying(10) | Kode cerita pengguna |
| id_kata_user_story | Serial | Identitas unik dari setiap kata |
| kata_user_story | character varying(100) | Kata dari user story |

Hasil dari proses ekstraksi kata dalam cerita pengguna ini akan disimpan kedalam tabel *user_story_kata* sehingga hasil yang diperoleh dari cerita pengguna pada tabel 4.3 akan diproses menggunakan metode *simpanKataUserStory* dan hasilnya berupa kumpulan kata-kata seperti tampak pada tabel 4.5

```
1  private void simpanKataUserStory(String sKode, String
2  sUS) {
3       StopList stopList = new StopList(theKoneksi);
4        stopList.initialize();
5        // Ganti semua tanda baca dengan  spasi dan
6  masukan        kedalam array string
7       String[] arrayKata = sUS.replace(".", "
8       ").replace(",", " ").split(" ");
9        for (int i = 0; i < arrayKata.length; i++) {
10       try {
11       //Inisialisasikan kelas Kata untuk setiap kata
12 dalam         array kata
13      Kata baru = new Kata(arrayKata[i], this.kamus);
14      /**
15      * method getBentukDasar,  menghilangkan
16      * akhiran terlebih dahulu-baru awalan , atau awalan
17      * dahulu-baru akhiran, sesuai RULE PRECEDENCE yg
18      * dinyatakan Jelita Asian.
19      * Penanganan stemming untuk kata ulang, dilakukan
20      * terakhir, setelah penghilangan awalan-akhiran
21      * terakhir, setelah penghilangan awalan-akhiran
22 gagal.
      if (baru.getBentukDasar(i).length() > 1) {
23      if (!stopList.isContain(baru.getBentukDasar(i))) {
24           theKoneksi.executeUpdate("INSERT INTO
25
      user_story_kata(id_user_story,kata_user_story)
26           VALUES ('" + sKode + "','" +
27                baru.getBentukDasar(i) + "')");
28           modelKataUserStory.addRow(new Object[]{sKode,
29                baru.getBentukDasar(i)});
30      }
31 }
32 } catch (SQLException ex) {
33      Logger.getLogger(ParsingJavaCode1.class.getName()).
34 log   (Level.SEVERE, null, ex);
35 } catch (IllegalStateException ex) {
36      Logger.getLogger(ParsingJavaCode1.class.getName()).
37 log   (Level.SEVERE, null, ex);
38 }
39 }
40 }
```

Gambar 4.2 Potongan Kode proses stemm dan menyimpan kata dari cerita pengguna

Tabel 4.5. Contoh  isi tabel *user_story_kata*

| id_user_story | id_user_story_kata | kata_user_story |
|---|---|---|
| P001-001 | 1 | sebagai |
| P001-001 | 2 | pengguna |
| P001-001 | 3 | ingin |
| P001-001 | 4 | milik |
| P001-001 | 5 | halaman |

| | | |
|---|---|---|
| P001-001 | 6 | login |
| P001-001 | 7 | dapat |
| P001-001 | 8 | Masuk |
| P001-001 | 9 | Dalam |
| P001-001 | 10 | Sistem |
| P001-001 | 11 | User |
| P001-001 | 12 | password |

### 4.2.2 Ekstraksi Kode Sumber

Proses ekstraksi kode sumber seperti telah dijelaskan pada gambar 3.3, proses dimulai dengan memilih lokasi atau folder dimana file kode sumber disimpan. Selanjutnya system akan mengumpulkan file yang berekstensi *.java* kedalam tabel *source_code* dengan struktur tabel seperti pada tabel 4.6.

Tabel 4.6 Tabel *source_code*

| Nama Kolom | Tipe data | Keterangan |
|---|---|---|
| id_project | character varying(10) | Kode proyek |
| id_source_code | character varying(10) | Kode Kode sumber |
| nama_file | Text | Path dan nama file kode sumber |

Selanjutnya system akan membaca seluruh isi folder yang dipilih dan menyimpan pada tabel *source_code* seperti contoh pada tabel 4.7. Contoh isi tabel *source_code*

Tabel 4.7 Contoh isi tabel *source_code*

| id_project | id_source_code | nama_file |
|---|---|---|
| P001 | P001-001 | /home/tesis/absensi/ReportParam.java |
| P001 | P001-002 | /home/tesis/absensi/crypto/FileEncryption.java |
| P001 | P001-003 | /home/tesis/absensi/crypto/CipherExample.java |
| P001 | P001-004 | /home/tesis/absensi/crypto/CryptoUtilsTest.java |
| P001 | P001-005 | /home/tesis/absensi/crypto/CryptoException.java |
| P001 | P001-006 | /home/tesis/absensi/crypto/CryptoUtils.java |
| P001 | P001-007 | /home/tesis/absensi/web/HariLiburController.java |
| P001 | P001-008 | /home/tesis/absensi/web/SKPDController.java |
| P001 | P001-009 | /home/tesis/absensi/web/ServiceController.java |
| P001 | P001-010 | /home/tesis/absensi/web/RoleController.java |

Proses pengkodean kode sumber seperti pada tabel 4.7 berfungsi untuk mempermudah proses otomatisasi kerunutan serta menyingkat nama file program menggunakan identitas kode sumber. Pada proses selanjutnya kode sumber yang tersinpan pada tabel tersebut akan diparsing satu persatu dengan bantuan *javaparser library* versi 1.0.11 (*javaparser-1.0.11.jar*). Pustaka *javaparser* sebagai kakas yang digunakan untuk melakukan parsing pada kelas, metode, komentar dan *attribute* sehingga dapat diidentifikasi nama kelas, nama konstruktor, nama metode, isi komentar dan nama *attribute*. Proses parsing kode sumber dapat dilihat pada gambar

4.3. Potongan motode parsing kode sumber.

```
1  public void parsingSourceCode() {
2     String fileName;
3     koneksi = new Koneksi();
4     this.kamus = new Kamus(this.koneksi);
5     this.kamus.initialize();
6     String nol = "000";
7     progressBar.setEnabled(true);
8     progressBar.setMaximum(modelFileSourceCode.getRowCount());
9     for (int i = 0; i < modelFileSourceCode.getRowCount(); i++)
10 {
11       FileInputStream in = null;
12        try {
13           sIDSC = nol.substring(0, 3 -
14 idSC.toString().length()) +
15    idSC.toString();
16           fileName = modelFileSourceCode.getValueAt(i,
17 0).toString();
18           final int ii = i;
19           in = new FileInputStream(fileName);
20           CompilationUnit cu = null;
21           // Mulai memparsing file
22            cu = JavaParser.parse(in);
23            // Memanggil metode untuk parsing class
24            ClassVisitor className = new ClassVisitor();
25            className.visit(cu, null);
26
27            //Memanggil metode untuk parsing classComment
28            CommentVisitor commentVisit = new CommentVisitor();
29            commentVisit.visit(cu, className.getClassName());
30
31            // Memanggil metode untuk parsing metode
32            MethodVisitor methodName = new MethodVisitor();
33            methodName.visit(cu, className.getClassName());
34
35           // Memanggil metode untuk parsing attribute
36            FieldVisitor fieldName = new FieldVisitor();
37            fieldName.visit(cu, className.getClassName());
38
```

```
39
                // Memanggil metode untuk parsing Constructor
40              ConstructorVisitor constructorName = new
41              ConstructorVisitor();
42               constructorName.visit(cu, className.getClassName());
43
44              doProgessBar(i + 1, fileName);
                modelFileSourceCode.setValueAt(true, i, 1);
45              idSC++;
46            } catch (FileNotFoundException ex) {
47               Logger.getLogger(SourceCodeParser.class.getName()).
48               log(Level.SEVERE, null, ex);
49            } catch (ParseException ex) {
50               Logger.getLogger(SourceCodeParser.class.getName()).
51               log(Level.SEVERE, null, ex);
52            } finally {
53               try {
54                  in.close();
55                  } catch (IOException ex) {
56                       Logger.getLogger(SourceCodeParser.class.
57                       getName()).log(Level.SEVERE, null, ex);
58                  }
59               }
60            }
61
62
```

Gambar 4.3 Potongan Kode proses parsing kode sumber

Pada gambar 4.3. ditunjukkan bagaimana metode *parsingSourceCode()* melakukan parsing pada setiap file kode sumber, setiap file akan dibaca dan diamati dengan memanggil masing masing metode yang bernama *visit*. Didalam setiap metode visit kata yang diterima baik nama kelas, metode, atrribut dan konstruktor akan diparsing kembali sesuai dengan aturan atau konvensi penamaan *java* seperti pada tabel 2.2. Potongan program untuk parsing sesuai aturan tersebut dapat dilihat pada gambar 4.4. Apabila ditemukan kata sambung maka kata tersebut akan dihilangkan atau tidak disimpan dalam tabel.

```
1   public List<String> hasilParsingJavaName() {
2           Koneksi theKoneksi;
3           theKoneksi = new Koneksi();
4           StopList kataSambung = new StopList(theKoneksi);
5           kataSambung.initialize();
6           List<String> returnKata = new ArrayList<String>();
7           if (jenisIdentifier.equalsIgnoreCase("CLASS")) {
8               returnKata = parsingClassName(isiIdentifier);
9           } else if
10  (jenisIdentifier.equalsIgnoreCase("METHOD")) {
11              returnKata = parsingMethod(isiIdentifier);
12          } else if
```

```
13  (jenisIdentifier.equalsIgnoreCase("ATTRIBUTE")) {
14              returnKata = parsingAttribute(isiIdentifier);
15          } else if
16  (jenisIdentifier.equalsIgnoreCase("METHOD")) {
17              returnKata = parsingMethod(isiIdentifier);
18          } else if
19  (jenisIdentifier.equalsIgnoreCase("COMMENT")) {
20              returnKata = parsingComment(isiIdentifier);
21          } else if
22  (jenisIdentifier.equalsIgnoreCase("CONSTRUCTOR")) {
23              returnKata = parsingClassName(isiIdentifier);
24          }
25          ArrayList<String> wordsToRemove = new
    ArrayList<String>();
26          for (String kata : returnKata) {
27              if (kataSambung.isContain(kata)) {
28                  wordsToRemove.add(kata);
29              }
30          }
31          for (String kata : wordsToRemove) {
32              returnKata.remove(returnKata.indexOf(kata));
33          }
34          wordsToRemove.clear();
35          return returnKata;
36      }
```

Gambar 4.4 Potongan Kode proses parsing konvensi penamaan java

Untuk metode *parsingComment()* memiliki isi metode yang berbeda dari yang lainnya, karena komentar biasanya berisi kalimat, sehingga masih diperlukan proses parsing, stemming sampai ditemukan bentuk kata dasarnya selanjutnya kata tersebut disimpan dalam tabel *source_code_kata* dengan struktur tabel seperti pada tabel.

Hasil dari metode *parsingComment()* berupa daftar kata-kata dasar yang disimpan pada tabel *source_code*, *id_source_code* akan diisi dengan kode dari setiap kode sumber yang berelasi dengan tabel *source_code*. *id_source_code_kata* diisi dengan angka yang secara otomatis akan dibuat oleh *postgreSQL*. Jenis akan diisi jenis dari kode sumber dapat berisi *method*, *class*, *constructor*, *attribute*, atau *comment.* Kata akan diisi dengan kata-kata dasar yang terkandung dari setiap *identifier* seperti contoh pada tabel 4.9. contoh isi tabel *source_code*.

Tabel 4.8 Tabel *source_code*

| Nama Kolom | Tipe data | Keterangan |
|---|---|---|
| id_source_code | character varying(10) | Indentitas Kode sumber |
| id_source_code_kata | Serial | Identitas unik kata kode sumber |
| nama_class | character varying(255) | Nama kelas |
| Jenis | character varying(255) | Jenis Kode Sumber |
| Kata | character varying(255) | Kata dalam kode sumber |

Tabel 4.9 contoh isi tabel *source_code*

| id_source_code | id_source_code_kata | nama_class | jenis | kata |
|---|---|---|---|---|
| P001-001 | 1 | ReportParam | Class | report |
| P001-002 | 2 | ReportParam | Class | Param |
| P001-003 | 3 | ReportParam | Method | Bulan |
| P001-004 | 4 | ReportParam | Method | Tahun |
| P001-005 | 5 | ReportParam | Method | Tanggal1 |
| P001-006 | 6 | ReportParam | Method | Tanggal2 |
| P001-007 | 7 | ReportParam | Comment | @author |
| P001-008 | 8 | ReportParam | Comment | faheem |

### 4.2.3 Membangun Kerunutan Hubungan

Hubungan kerunutan antara tabel cerita pengguna dan tabel kode sumber menggunakan string matching dengan algoritma *trigram* yang telah diintegrasikan dengan postgreSQL (pg_trgrm). Modul *pg_trgm* menyediakan fungsi dan operator untuk menentukan kesamaan teks alfanumerik berdasarkan pencocokan trigram, serta kelas Operator indeks yang mendukung pencarian string yang sama secara cepat. Pada *pg_trgm* karakter non-kata (non-alphanumerics) diabaikan ketika mengekstrak trigram dari string. Setiap kata dianggap memiliki dua ruang diawali dan satu ruang akhiran ketika menentukan set trigram yang terkandung dalam string. Misalnya, himpunan trigram dalam string "cat" adalah "c", "ca", "cat", dan "at". Himpunan trigram dalam string "foo | bar" adalah "f", "fo", "foo", "oo", "b", "ba", "bar", dan "ar". (The PostgreSQL , 1996-2016)

*Pg_trgm* memiliki beberapa fungsi dan operator yang dapat digunakan seperti pada tabel 4.10. Fungsi *pg_trigram* dan Tabel 4.11 Operator *pg_trigram*

Tabel 4.10 Fungsi pg_trigram

| Fungsi | Keluaran | Keterangan |
|---|---|---|
| *similarity(text, text)* | *real* | Nilai keluaran berupa angka antara 0 sampai 1, 0 menunjukkan kedua argument tidak mirip sama sekali dan 1 menunjukkan tingat kemiripan 100% |
| *show_trgm(text)* | *text[]* | Keluaran dari fungsi berupa *array string* dari trigrams. (Dalam prakteknya ini jarang berguna kecuali untuk *debugging*) |
| *show_limit()* | *real* | Nilai keluaran berupa angka yang menunjukan ambang batas yang digunakan operator %. Merupakan ambang batas yang menentukan apakah antara dua kata yang dibandingkan cukup mirip atau tidak.. |
| *set_limit(real)* | *real* | Menentukan ambang batas yang akan digunakan oleh operator %, dan akan mempengaruhi nilai keluaran fungsi *show_limit()*. |

Tabel 4.11 Operator *pg_trigram*

| Fungsi | Keluaran | Keterangan |
|---|---|---|
| *text % text* | *boolean* | Nilai keluaran *true* jika kedua argumen dianggap mirip sesuai ambang batas dari fungsi *show_limit()* |
| *text <-> text* | *real* | Nilai Keluaran angka |

Pada penelitian ini yang digunakan adalah fungsi *similarity(text, text)* yang akan digunakan untuk membandingkan dua string atau kata dan diambil nilai keluarannya berupa bilangan *real*, sebagai gambaran akan diberikan contoh pada gambar 4.5. Contoh query penggunaan fungsi *similarity(text, text)*.

Dari gambar 4.5. dapat dilihat antara “tanggal” dan “tanggal” memiliki nilai kemiripan 1, “tanggal” dan “tanggal1” memiliki nilai kemiripan 0.7, “tanggal” dan “tgl” memiliki nilai kemiripan 0.0909091 serta “tanggal” dan “date” memiliki nilai kemiripan 0 yang artinya tidak ada satu karakterpun yang mirip antara keduanya.

```
tesis=# SELECT similarity('tanggal','tanggal');
 similarity
------------
          1
(1 row)

tesis=# SELECT similarity('tanggal','tanggal1');
 similarity
------------
        0.7
(1 row)

tesis=# SELECT similarity('tanggal','tgl');
 similarity
------------
  0.0909091
(1 row)

tesis=# SELECT similarity('tanggal','date');
 similarity
------------
          0
(1 row)
```

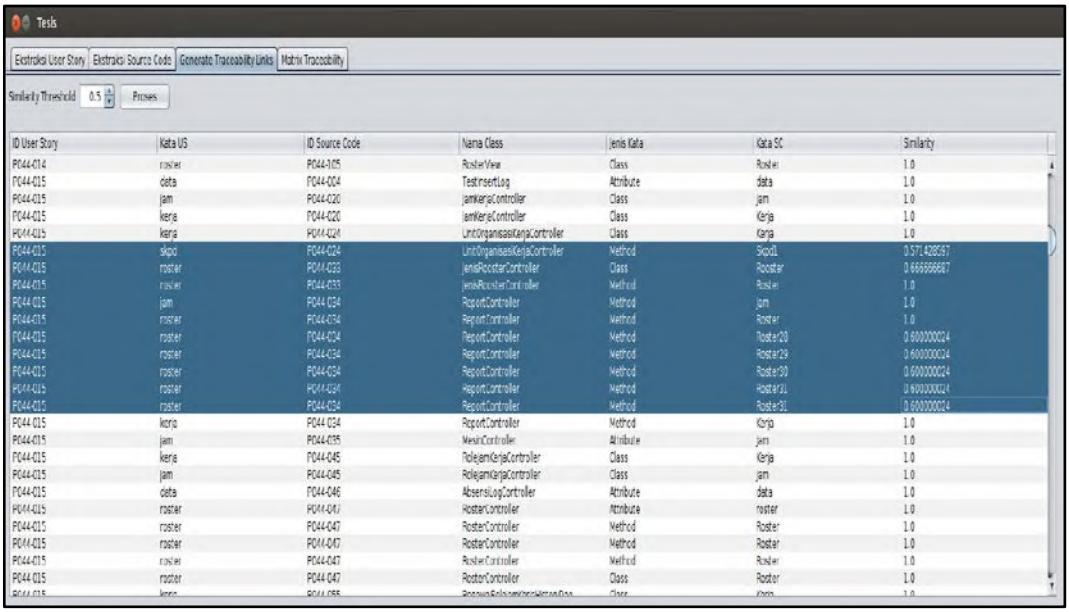
Gambar 4.5 Contoh penggunaan fungsi similarity

Indentifikasi adanya hubungan antara cerita pengguna dengan kode sumber menggunakan query yang nantinya dapat disesuaikan ambang batasnya sebagai bahan percobaa, sehingga dapat ditentukan nilai kemiripan antara kata yang terkandung dalam cerita pengguna dengan kata yang terkandung dalam kode sumber. *Query* yang digunakan dalam system ini dapat dilihat pada gambar 4.6.Potongan program mencari kemiripan kata cerita pengguna dengan kode sumber.

```
1  private void prosesTraceability() {
2    modelSimilarity = (DefaultTableModel)
3    tableSimilarity.getModel();
4    modelSimilarity.setRowCount(0);
5    try {
6        String queryTL = "SELECT "
         + "id_source_code,nama_class,jenis,kata as “
         + "kata_source_code,”
         + "id_user_story, kata_user_story, "
         + "similarity(kata,kata_user_story) from "
         + "source_code_kata,”
         + "user_story_kata "
         + "WHERE id_source_code like '" + idProject + "%' “
         + "and id_user_story like '" + idProject +"%'  "
         + "and similarity(kata,kata_user_story) >"
         + rTreshHold.getValue().toString()
         + " ORDER BY id_user_story, id_source_code";
```

```
7         ResultSet rsTL = theKoneksi.executeSelect(queryTL);
8         while (rsTL.next()) {
             modelSimilarity.addRow(new
             Object[]{rsTL.getString("id_user_story"),
             rsTL.getString("kata_user_story"),
             rsTL.getString("id_source_code"),
             rsTL.getString("nama_class"),
             rsTL.getString("jenis"),
             rsTL.getString("kata_source_code"),
             rsTL.getDouble("similarity")
9             });
10        }
11    } catch (SQLException ex) {
12       Logger.getLogger(ParsingJavaCode1.class.getName()).
13       log(Level.SEVERE, null, ex);
14    } catch (IllegalStateException ex) {
15       Logger.getLogger(ParsingJavaCode1.class.getName()).
16       log(Level.SEVERE, null, ex);
17    }
18 }
19
```

Gambar 4.6. Potongan program mencari kemiripan kata cerita pengguna dengan kode sumber




Gambar 4.7. Hasil kemiripan kata cerita pengguna dengan kode sumber.

Indentifikasi adanya hubungan antara cerita pengguna dengan kode sumber menggunakan query yang nantinya dapat disesuaikan ambang batasnya sebagai bahan percobaan, sehingga dapat ditentukan nilai kemiripan antara kata yang terkandung dalam cerita pengguna dengan kata yang terkandung dalam kode sumber. *Query* yang digunakan dalam system ini dapat dilihat pada gambar 4.6.Potongan

program mencari kemiripan kata cerita pengguna dengan kode sumber. Hasil dari eksekusi program tersebut seperti tampak pada gambar 4.7. Hasil pencarian kemiripan kata cerita pengguna dank ode sumber

### 4.2.4 Membentuk Matrik Kerunutan Hubungan

Matrik hubungan kerunutan antara tabel cerita pengguna dan tabel kode sumber dibentuk berdasarkan hasil pencarian kemiripan kata antara keduanya. Apabila terdapat kata-kata yang memiliki kemiripan dengan ambang batas yang telah ditentukan maka dianggap memiliki hubungan.kerunutan. Agar lebih optimal ditambahkan perbandingan jumlah kata yang ditemukan pada kode sumber dengan cerita pengguna sebagai ambang batasnya.

Sistem ini menggunakan query untuk membentu matrik kerunutan hubungan dibentuk seperti tampak pada gambar 4.8. Query Pembentukan Matrik Kerunutan. Pada queri tersebut tampak terdapat 4 parameter yaitu:

- p_kode_SC

  Parameter ini diisi dengan kode dari kode sumber misalnya: “P001-023”
- p_kode_US

  Parameter ini diisi dengan kode dari cerita pengguna misalnya: “P001-001”
- p_similarity

  Parameter ini diisi dengan besaran nilai ambang batas kemiripan kata misalnya: 0.5
- p_prosen_kata

  Parameter ini diisi dengan prosentase jumlah kata yang terkandung dalam kode sumber dibandingkan dengan jumlah kata yang terkandung dalam cerita pengguna.

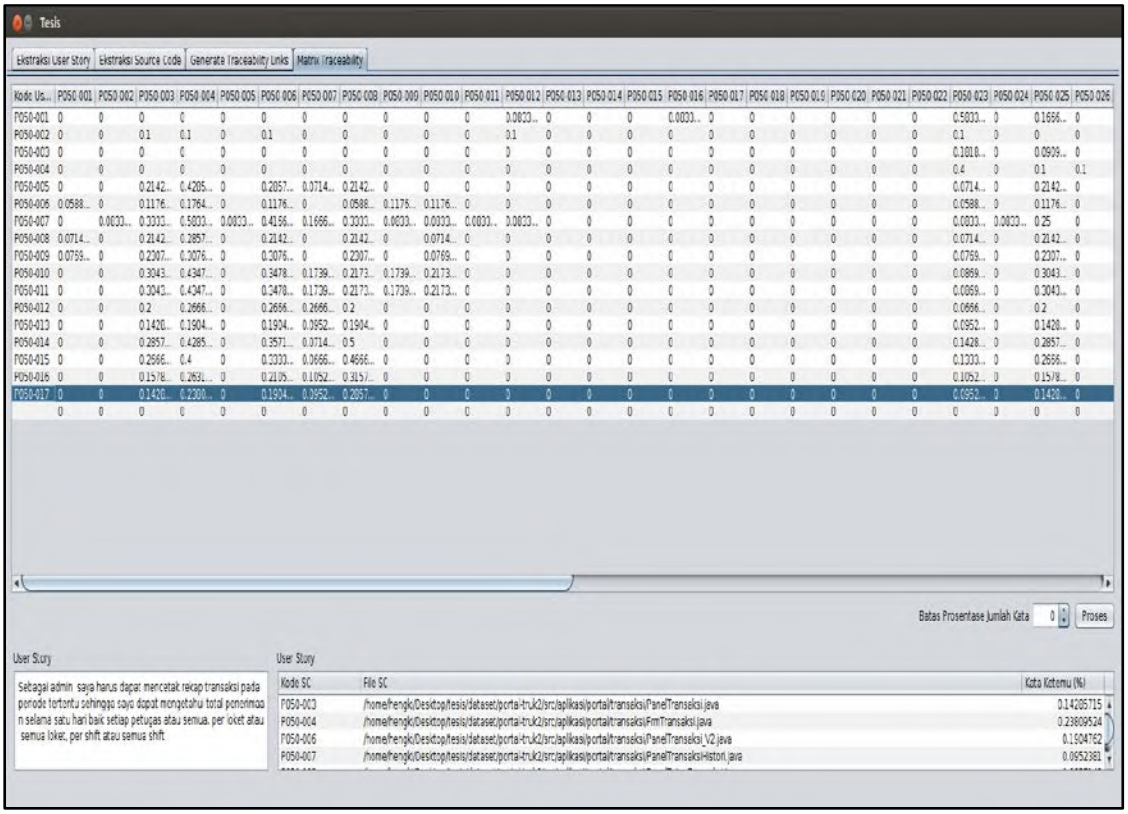
```
QUERY =#
SELECT sl.id_user_story,
   id_source_code,
   count(distinct kata_source_code) jml_kata_sc
   count(distinct us.kata_user_story) as jml_kata_us,
   count(distinct kata_source_code)::real/count(distinct
   us.kata_user_story)::real as persen_kata
   FROM (SELECT id_source_code,nama_class,jenis,kata as
       kata_source_code, id_user_story, kata_user_story,
```

```
      similarity(kata,kata_user_story)
      FROM source_code_kata, user_story_kata
         WHERE id_source_code = p_kode_SC
            and id_user_story = P_kode_US
            and similarity(kata,kata_user_story) >=
            P_Similarity
         ORDER BY id_user_story, id_source_code
  ) as sl
INNER JOIN user_story_kata us on
  sl.id_user_story = us.id_user_story
GROUP BY sl.id_user_story,id_source_code
HAVING count(distinct
   kata_source_code)::real/count(distinct
   us.kata_user_story)::real > P_Prosen_Kata
```

```
Hasil Query
id_user_story    id_source_code    jml_kata_sc
                 jml_kata_us       persen_kata
-------------------------------------------------
"P001-001"       "P001-023"        7              12
                 0.583333
```

Gambar 4.8.Query Pembentukan Matrik Kerunutan

Hasil dari query ini seperti tampak pada gambar 4.8 yang menunjukkan nilai 0.583333 artinya jumlah kata yang mirip dari kode sumber yang terkandung dalam cerita pengguna sebesar 58% lebih. Query tersebut dieksekusi pada setiap baris dan kolom dar matrix kerunutan hubungan yang hasilnya dapat dilihat pada gambar 4.9. Matrik kerunutan hubungan.




Gambar 4.9. Matrik kerunutan hubungan.

## 4.3 Skenario Pengujian

Pengujian terhadap metode yang ditawarkan pada penelitian ini dilakukan dengan mengimplementasikan alur kerja yang sudah diusulkan dan dijelaskan pada metodologi. Alur kerja tersebut diimplementasikan pada dataset yang akan menjadi bahan uji coba. Dataset yang digunakan untuk uji coba dibagi menjadi tiga kategori yaitu ukuran kecil, sedang dan besar.

Pengujian dilakukan dengan membandingkan hasil otomatisasi identifikasi kerunutuan dengan hasil dari identifikasi yang dilakukan oleh pengembang masing-masing dataset. Pengembang yang dipilih untuk melakukan identifikasi adalah orang-orang yang terlibat mulai dari pendefinisian kebutuhan sampai dengan programmer yang terlibat pembuatan kode sumber dari masing-masing dataset.

Proses identifikasi oleh sistem dilakukan secara otomatis dengan mencoba memberikan beberapa nilai threshold yang berbeda, hasil dari masing-masing ujicoba akan disimpan dalam database. Hasil identifikasi system tersebut selanjutnya dianalisis dengan cara membandingkan dengan hasil identifikasi oleh tim pengembang. Metode pembandingan antara hasil identifikasi oleh system dengan

hasil identifikasi pengembang dengan mengukur nilai *precission, recall dan accuracy* dari masing-masing dataset.

## 4.4 Analisis Hasil

Pengujian dilakukan pada ketiga dataset dari tiga sistem aplikasi yaitu smartPortal, smartAbsensi, dan smartTravel. Masing-masing memiliki jumlah cerita pengguna dan kode sumber yang beragam, jumlah total cerita pengguna sebanyak 144 dan jumlah kode sumber sebanyak 435.

### 4.4.1 Pengujian smartPortal

Aplikasi ini digunakan untuk memungut retribusi truk pengangkut galian C. Pada aplikasi ini digunakan IP camera sebagai alat untuk menangkap gambar truk yang selanjutnya diolah menggunakan HAAR training untuk mengetahui klasifikasi kendaraan yang memasuki pos penarikan retribusi. Aplikasi ini dibangun dengan bahasa pemrograman *java* desktop dengan *swing* yang selanjutnya disebut smartPortal. *SmartPortal* terdiri dari 17 cerita pengguna dan 52 file kode sumber. Daftar cerita pengguna dapat dilihat pada tabel 4.12. Tabel cerita pengguna smartPortal. Daftar kode sumber dapat dilihat pada tabel 4.13. Tabel file kode sumber.

Tabel 4.12. Tabel cerita pengguna smartPortal

| Kode Cerita Pengguna | Cerita Pengguna |
|---|---|
| US-001 | Sebagai pengguna saya ingin memiliki halaman login sehingga saya dapat masuk kedalam sistem menggunakan user dan password. |
| US-002 | Sebagai pengguna saya ingin memiliki tombol logout sehingga saya dapat masuk keluar dari sistem. |
| US-003 | Sebagai pengguna saya harus dapat merubah password saya sehingga saya dapat masuk kedalam sistem dengan password yang baru. |
| US-004 | Sebagai superuser saya harus dapat mengelola user sehingga saya dapat menambah user, merubah user, mencari user, dan menghapus user. |
| US-005 | Sebagai petugas loket saya harus dapat menerima pembayaran portal kendaraan sesuai dengan kelasnya sehingga saya dapat menyimpan hasil transaksi. |
| US-006 | Sebagai petugas loket saya harus dapat menerima informasi jenis kendaraan secara otomatis dari deteksi kamera cctv sehingga saya dapat lebih mudah menentukan jenis kendaraan. |
| US-007 | Sebagai petugas loket saya harus dapat mencetak nota pembayaran transaksi sehingga sopir kendaraan dapat menerima nota kendaraan. |
| US-008 | Sebagai petugas loket saya harus dapat membuka dan menutup portal truk sehingga portal dapat terbuka dan tertutup secara otomatis setelah melakukan pembayaran. |
| US-009 | Sebagai petugas loket saya harus dapat menahan portal truk melalui aplikasi sehingga portal dapat terbuka sementara sampai saya menutup kembali. |
| US-010 | Sebagai koordinator loket saya harus dapat melihat history transaksi pada periode waktu tertentu sehingga saya dapat mengetahui nomor faktur, jenis transaksi, jenis armada, tarif, jumlah pembayaran, nama petugas, shift, nomor loket, dan pembatalannya. |
| US-011 | Sebagai koordinator loket saya harus dapat mencetak history transaksi pada periode waktu tertentu sehingga saya dapat mengetahui nomor faktur, jenis transaksi, jenis armada, tarif, jumlah pembayaran, nama petugas, shift, nomor |

| | |
|---|---|
| | loket, dan pembatalannya. |
| US-012 | Sebagai koordinator loket saya harus dapat melihat history transaksi pada periode waktu tertentu sehingga saya dapat membatalkan transaksi bila terjadi kesalahan. |
| US-013 | Sebagai koordinator loket saya harus dapat mencetak foto kendaraan yang masuk portal pada saat transaksi pada periode waktu tertentu sehingga saya dapat memiliki foto tersebut sebagai bukti apabila terjadi kecurangan. |
| US-014 | Sebagai petugas loket saya harus dapat melakukan penutupan transaksi setiap shift sehingga saya dapat mengetahui total penerimaan selama satu shift setiap petugas. |
| US-015 | Sebagai petugas loket saya harus dapat mencetak hasil penutupan transaksi setiap shift sehingga saya dapat mengetahui total penerimaan selama satu shift setiap petugas. |
| US-016 | Sebagai admin saya harus dapat mencetak rekap harian transaksi sehingga saya dapat mengetahui total penerimaan selama satu hari baik setiap petugas atau semua, per loket atau semua loket, per shift atau semua shift. |
| US-017 | Sebagai admin saya harus dapat mencetak rekap transaksi pada periode tertentu sehingga saya dapat mengetahui total penerimaan selama satu hari baik setiap petugas atau semua, per loket atau semua loket, per shift atau semua shift. |

Tabel 4.13. Tabel kode sumber smartPortal

| Kode Kode Sumber | File Kode Sumber |
|---|---|
| SC-001 | /portal-truk2/src/aplikasi/detector/DetekTrukStream.java |
| SC-002 | /portal-truk2/src/aplikasi/portal/transaksi/DlgKonfirmasi.java |
| SC-003 | /portal-truk2/src/aplikasi/portal/transaksi/PanelTransaksi.java |
| SC-004 | /portal-truk2/src/aplikasi/portal/transaksi/FrmTransaksi.java |
| SC-005 | /portal-truk2/src/aplikasi/portal/transaksi/DlgImageTrx.java |
| SC-006 | /portal-truk2/src/aplikasi/portal/transaksi/PanelTransaksi_V2.java |
| SC-007 | /portal-truk2/src/aplikasi/portal/transaksi/PanelTransaksiHistori.java |
| SC-008 | /portal-truk2/src/aplikasi/portal/transaksi/PanelTutupTransaksi.java |
| SC-009 | /portal-truk2/src/aplikasi/portal/master/MasterJenisArmada.java |
| SC-010 | /portal-truk2/src/aplikasi/portal/master/PanelSettingTarifJenisTruk.java |
| SC-011 | /portal-truk2/src/aplikasi/portal/component/NewJFrame.java |
| SC-012 | /portal-truk2/src/aplikasi/portal/component/FungsiUmum.java |
| SC-013 | /portal-truk2/src/aplikasi/portal/component/MenuAuth.java |
| SC-014 | /portal-truk2/src/aplikasi/portal/component/OSValidator.java |
| SC-015 | /portal-truk2/src/aplikasi/portal/component/ColumnResizer.java |
| SC-016 | /portal-truk2/src/aplikasi/portal/component/panelLogin.java |
| SC-017 | /portal-truk2/src/aplikasi/portal/component/JTabbedPaneImage.java |
| SC-018 | /portal-truk2/src/aplikasi/portal/component/ButtonTabComponent.java |
| SC-019 | /portal-truk2/src/aplikasi/portal/component/MyRowRenderer.java |
| SC-020 | /portal-truk2/src/aplikasi/portal/component/PanelHeader.java |
| SC-021 | /portal-truk2/src/aplikasi/portal/component/MyPanel.java |
| SC-022 | /portal-truk2/src/aplikasi/portal/KoneksiDB.java |
| SC-023 | /portal-truk2/src/aplikasi/portal/form/FormLogin.java |
| SC-024 | /portal-truk2/src/aplikasi/portal/form/Menu.java |
| SC-025 | /portal-truk2/src/aplikasi/portal/form/FormUtama.java |
| SC-026 | /portal-truk2/src/aplikasi/portal/form/HapusTabDiJTabbedPane.java |
| SC-027 | /portal-truk2/src/aplikasi/portal/form/DlgMasterJenisArmada.java |
| SC-028 | /portal-truk2/src/aplikasi/portal/form/Tarif.java |
| SC-029 | /portal-truk2/src/aplikasi/portal/form/DlgUbahTransaksi.java |
| SC-030 | /portal-truk2/src/aplikasi/portal/form/UserManagementPanel.java |
| SC-031 | /portal-truk2/src/aplikasi/portal/form/PanelReport.java |
| SC-032 | /portal-truk2/src/aplikasi/portal/form/DlgUbahPassword.java |
| SC-033 | /portal-truk2/src/aplikasi/portal/form/Deteksi.java |
| SC-034 | /portal-truk2/src/aplikasi/portal/model/SettingTarif.java |
| SC-035 | /portal-truk2/src/aplikasi/portal/model/RoleUser.java |
| SC-036 | /portal-truk2/src/aplikasi/portal/model/Transaksi.java |
| SC-037 | /portal-truk2/src/aplikasi/portal/model/JenisTransaksi.java |
| SC-038 | /portal-truk2/src/aplikasi/portal/model/SettingTarifDetail.java |
| SC-039 | /portal-truk2/src/aplikasi/portal/model/UserModel.java |
| SC-040 | /portal-truk2/src/aplikasi/portal/dao/JenisArmadaDao.java |
| SC-041 | /portal-truk2/src/aplikasi/portal/dao/UserDao.java |
| SC-042 | /portal-truk2/src/aplikasi/portal/dao/SettingTarifDao.java |
| SC-043 | /portal-truk2/src/aplikasi/portal/dao/JenisTransaksiDao.java |
| SC-044 | /portal-truk2/src/aplikasi/portal/dao/TransaksiDao.java |
| SC-045 | /portal-truk2/src/aplikasi/portal/service/TangkapGambar.java |
| SC-046 | /portal-truk2/src/aplikasi/portal/service/ReportService.java |
| SC-047 | /portal-truk2/src/aplikasi/portal/service/ESCPrinter.java |
| SC-048 | /portal-truk2/src/aplikasi/portal/service/PrintTrxService.java |

| | |
|---|---|
| SC-049 | /portal-truk2/src/aplikasi/portal/Main.java |
| SC-050 | /portal-truk2/src/aplikasi/portal/JenisArmada.java |
| SC-051 | /portal-truk2/src/aplikasi/barriergate/BarrierProtocol.java |
| SC-052 | /portal-truk2/src/aplikasi/barriergate/SerialRXTX.java |

Pengujian kerunutan hubungan *smartPortal* akan dibandingkan dengan matrik kerunutan hubungan yang telah dibuat oleh tim pengembang seperti tabel 4.14. Tabel matrik kerunutan hubungan dari pengembang, pada tabel tersebut terlihat hubungan antara kode sumber dengan cerita pengguna ditandai dengan cell yang bernilai 1. Total hubungan kerunutan yang diberikan pengembang untuk smartPortal adalah 47, terdapat 21 cerita pengguna yang tidak memiliki hubungan kerunutan manapun, hal ini akan mempengaruhi nilai presisi dari ujicoba nantinya.

Tabel 4.14. Tabel matrik kerunutan hubungan dari pengembang

| Matrik Kerunutan Pengembang | | Cerita Pengguna | | | | | | | | | | | | | | | | | TOT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | US-001 | US-002 | US-003 | US-004 | US-005 | US-006 | US-007 | US-008 | US-009 | US-010 | US-011 | US-012 | US-013 | US-014 | US-015 | US-016 | US-017 | |
| Kode Sumber | SC-001 | | | | | | 1 | | | | | | | | | | | | 1 |
| | SC-002 | | | | | | | | | | | | | | | | | | 0 |
| | SC-003 | | | | | | | | | | | | | | | | | | 0 |
| | SC-004 | | | | | 1 | | | | | | | | | | | | | 1 |
| | SC-005 | | | | | 1 | | | | | | | | 1 | | | | | 2 |
| | SC-006 | | | | | 1 | 1 | 1 | 1 | 1 | | | | | | | | | 5 |
| | SC-007 | | | | | | | 1 | | | 1 | 1 | 1 | 1 | | | | | 5 |
| | SC-008 | | | | | | | | | | | | | | 1 | 1 | | | 2 |
| | SC-009 | | | | | | | | | | | | | | | | | | 0 |
| | SC-010 | | | | | | | | | | | | | | | | | | 0 |
| | SC-011 | | | | | | | | | | | | | | | | | | 0 |
| | SC-012 | | | | | | | | | | | | | | | | 1 | 1 | 2 |
| | SC-013 | | | | | | | | | | | | | | | | | | 0 |
| | SC-014 | | | | | | | | | | | | | | | | | | 0 |
| | SC-015 | | | | | | | | | | | | | | | | | | 0 |
| | SC-016 | 1 | | | | | | | | | | | | | | | | | 1 |
| | SC-017 | | | | | | | | | | | | | | | | | | 0 |
| | SC-018 | | | | | | | | | | | | | | | | | | 0 |
| | SC-019 | | | | | | | | | | | | | | | | | | 0 |
| | SC-020 | | | | | | | | | | | | | | | | | | 0 |
| | SC-021 | | | | | | | | | | | | | | | | | | 0 |
| | SC-022 | | | | | | | | | | | | | | | | | | 0 |
| | SC-023 | 1 | | | | | | | | | | | | | | | | | 1 |
| | SC-024 | | | | | | | | | | | | | | | | | | 0 |
| | SC-025 | | 1 | | | | | | | | | | | | | | | | 1 |
| | SC-026 | | | | | | | | | | | | | | | | | | 0 |

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SC-027 | | | | | | | | | | | | | | | | | | 0 |
| | SC-028 | | | | | | | | | | | | | | | | | | 0 |
| | SC-029 | | | | | | | | | | | | 1 | | | | | | 1 |
| | SC-030 | | | | 1 | | | | | | | | | | | | | | 1 |
| | SC-031 | | | | | | | | | | | | | | | | 1 | 1 | 2 |
| | SC-032 | | | 1 | | | | | | | | | | | | | | | 1 |
| | SC-033 | | | | | | 1 | | | | | | | | | | | | 1 |
| | SC-034 | | | | | | | | | | | | | | | | | | 0 |
| | SC-035 | | | | 1 | | | | | | | | | | | | | | 1 |
| | SC-036 | | | | | 1 | | | | | | | 1 | | | | | | 2 |
| | SC-037 | | | | | | | | | | | | | | | | | | 0 |
| | SC-038 | | | | | | | | | | | | | | | | | | 0 |
| | SC-039 | | | | 1 | | | | | | | | | | | | | | 1 |
| | SC-040 | | | | | | | | | | | | | | | | | | 0 |
| | SC-041 | | | | 1 | | | | | | | | | | | | | | 1 |
| | SC-042 | | | | | | | | | | | | | | | | | | 0 |
| | SC-043 | | | | | | | | | | | | | | | | | | 0 |
| | SC-044 | | | | | 1 | | | | | | | 1 | | | | | | 2 |
| | SC-045 | | | | | | 1 | | | | | | | | | | | | 1 |
| | SC-046 | | | | | | | 1 | | | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 8 |
| | SC-047 | | | | | | | 1 | | | | | | | | | | | 1 |
| | SC-048 | | | | | | | 1 | | | | | | | | | | | 1 |
| | SC-049 | | | | | | | | | | | | | | | | | | 0 |
| | SC-050 | | | | | | | | | | | | | | | | | | 0 |
| | SC-051 | | | | | | | | | | | | | | | | | | 0 |
| | SC-052 | | | | | | | | 1 | 1 | | | | | | | | | 2 |
| TOT | | 2 | 1 | 1 | 4 | 5 | 4 | 5 | 2 | 2 | 2 | 1 | 5 | 3 | 2 | 2 | 3 | 3 | 47 |

Untuk mempermudah penjelasan penulis memberikan salah satu contoh hasil matrik kerunutan hubungan yang dihasilkan system dengan *threshold simalirity* sebesar 0.7 dan *threshold* jumlah kata dalam kode sumber dibanding dengan jumlah kata dalam cerita pengguna sebesar 0.25. Pada hasil data uji yang lain tidak ditampilkan secara detail dengan pertimbangan banyaknya jumlah data. Tabel 4.15. menunjukkan matrik kerunutan hubungan dari system.

Tabel 4.15. Tabel matrik kerunutan hubungan dari sistem

| | | Cerita Pengguna | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 001 | 002 | 003 | 004 | 005 | 006 | 007 | 008 | 009 | 010 | 011 | 012 | 013 | 014 | 015 | 016 | 017 |
| Kode Sumber | 001 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 002 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 003 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,26 | 0,26 | 0,00 | 0,00 | 0,29 | 0,27 | 0,00 | 0,00 |
| | 004 | 0,00 | 0,00 | 0,00 | 0,00 | 0,36 | 0,00 | 0,42 | 0,29 | 0,31 | 0,35 | 0,35 | 0,00 | 0,00 | 0,36 | 0,33 | 0,00 | 0,00 |
| | 005 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 006 | 0,00 | 0,00 | 0,00 | 0,00 | 0,29 | 0,00 | 0,33 | 0,00 | 0,31 | 0,30 | 0,30 | 0,27 | 0,00 | 0,36 | 0,33 | 0,00 | 0,00 |
| | 007 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 008 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,50 | 0,47 | 0,32 | 0,29 |
| | 009 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 010 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,26 | 0,26 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 011 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 012 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 013 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 014 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 015 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 016 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 017 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 018 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 019 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 020 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 021 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 022 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 023 | 0,42 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 024 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 025 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,26 | 0,26 | 0,00 | 0,00 | 0,29 | 0,27 | 0,00 | 0,00 |
| | 026 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 027 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 028 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 029 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,35 | 0,35 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 030 | 0,33 | 0,00 | 0,00 | 0,50 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 031 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 032 | 0,00 | 0,00 | 0,27 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 033 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 034 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 035 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 036 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 037 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 038 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 039 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 040 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 041 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 042 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 043 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 044 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 045 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 046 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 047 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 048 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 049 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 050 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 051 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | 052 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |

Metode evaluasi sistem seperti telah dijelaskan pada bagian 2.10, maka setiap cell antara cerita pengguna dengan kode sumber hasil dari system dengan threshold tertentu (tabel 4.15) akan dibandingkan dengan hasil identifikasi dari pengembang (tabel 4.14) dengan aturan seperti pada gambar 4.10. Aturan evaluasi presisi dan recall.

```
If cell_matrik_dev == 1 && cell_matrik_sys >0
  Print "TP"
else
  If cell_matrik_dev  == 1 && cell_matrik_sys == 0
     Print "FN"
  else
     If cell_matrik_dev !=1 && cell_matrik_sys > 0
        Print "FP"
     else
        If cell_matrik_dev ==1 && cell_matrik_sys == 0
           Print "FN"
        else
        Print "TN"
```

Gambar 4.10. Aturan evaluasi presisi dan recall.

Pada gambar 4.10 jika hasil dari matrik pengembang bernilai 1 dan matrik dari system bernilai lebih dari 0 maka hasilnya adalah *true positive(TP)* artinya terdapat kesamaan antara hasil identifikasi pengembang dengan identifikasi system. Pada kondisi kedua jika hasil identifikasi pengembang bernilai 1 sedangkan hasil identifikasi system sama dengan 0 maka hasilnya adalah *false positive (FP)* artinya system tidak berhasil mengidentifikasi hubungan kerunutan. Pada kondisi ketiga jika hasil identifikasi pengembang tidak sama dengan 1 sedangkan hasil identifikasi system lebih dari 0 maka hasilnya adalah *false Negative (FN)* artinya system salah mengidentifikasi hubungan kerunutan yang tidak seharusnya teridentifikasi. Kondisi terakhir adalah apabila hasil identifikasi pengembang tidak sama dengan 1 dan hasil identifikasi system sama dengan 0 maka hasilnya adalah *true negative (TN)* artinya antara pengembang dan system terdapat kesamaan yaitu tidak ada hubungan kerunutan.

Berdasarkan aturan tersebut maka akan dihasilkan tabel 4.16. Tabel presisi dan recall hasil perbandingan identifikasi pengembang dan sistem. Dari tabel 4.16, perhitungan dilanjutkan dengan menentukan nilai presisi dan recall dari setiap cerita pengguna dengan hasil rata-rata presisi dan recall setiap percobaan seperti pada tabel 4.17. Pada percobaan selanjutnya tabel 4.17 inilah yang akan disajikan per percobaan.

Tabel 4.16. Perbandingan identifikasi pengembang dan sistem.

| | | Cerita Pengguna | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | US-001 | US-002 | US-003 | US-004 | US-005 | US-006 | US-007 | US-008 | US-009 | US-010 | US-011 | US-012 | US-013 | US-014 | US-015 | US-016 | US-017 |
| Kode Sumber | SC-001 | TN | TN | TN | TN | TN | FN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-002 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-003 | TN | TN | TN | TN | TN | TN | TN | TN | TN | FP | FP | TN | TN | FP | FP | TN | TN |
| | SC-004 | TN | TN | TN | TN | TP | TN | FP | FP | FP | FP | FP | TN | TN | FP | FP | TN | TN |
| | SC-005 | TN | TN | TN | TN | FN | TN | TN | TN | TN | TN | TN | TN | FN | TN | TN | TN | TN |
| | SC-006 | TN | TN | TN | TN | TP | FN | TP | FN | TP | FP | FP | FP | TN | FP | FP | TN | TN |
| | SC-007 | TN | TN | TN | TN | TN | TN | FN | TN | TN | FN | FN | FN | FN | TN | TN | TN | TN |
| | SC-008 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TP | TP | FP | FP |
| | SC-009 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-010 | TN | TN | TN | TN | TN | TN | TN | TN | TN | FP | FP | TN | TN | TN | TN | TN | TN |
| | SC-011 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-012 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | FN | FN |
| | SC-013 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-014 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-015 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-016 | FN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-017 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-018 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-019 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-020 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-021 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-022 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-023 | TP | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-024 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-025 | TN | FN | TN | TN | TN | TN | TN | TN | TN | FP | FP | TN | TN | FP | FP | TN | TN |
| | SC-026 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-027 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-028 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-029 | TN | TN | TN | TN | TN | TN | TN | TN | TN | FP | FP | FN | TN | TN | TN | TN | TN |
| | SC-030 | FP | TN | TN | TP | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-031 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | FN | FN |
| | SC-032 | TN | TN | TP | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-033 | TN | TN | TN | TN | TN | FN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-034 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-035 | TN | TN | TN | FN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-036 | TN | TN | TN | TN | FN | TN | TN | TN | TN | TN | TN | FN | TN | TN | TN | TN | TN |
| | SC-037 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-038 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-039 | TN | TN | TN | FN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-040 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-041 | TN | TN | TN | FN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-042 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-043 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-044 | TN | TN | TN | TN | FN | TN | TN | TN | TN | TN | TN | FN | TN | TN | TN | TN | TN |
| | SC-045 | TN | TN | TN | TN | TN | FN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-046 | TN | TN | TN | TN | TN | TN | FN | TN | TN | FN | TN | FN | FN | FN | FN | FN | FN |
| | SC-047 | TN | TN | TN | TN | TN | TN | FN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-048 | TN | TN | TN | TN | TN | TN | FN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-049 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-050 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-051 | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN | TN |
| | SC-052 | TN | TN | TN | TN | TN | TN | TN | FN | FN | TN | TN | TN | TN | TN | TN | TN | TN |

*Keterangan:*

| | |
|---|---|
| TP | True Positive |
| FP | False Positive |
| TN | True Negative |
| FN | False Negative |

Bedasarkan tabel 4.14, 4.15 dan 4.16. dihasilkan perhitungan seperti pada tabel 4.17 yaitu jumlah identifikasi pengembang, jumlah identifikasi system, jumlah masing-masing nilai TP,FP,FN, dan TN, nilai presisi dan recall serta rata-rata nilai presisi dan recall pada setiap percobaan.

Tabel 4.17. Hasil precission recall setiap cerita pengguna

| | | Jml Identifikasi | | Jumlah Perbandingan | | | | Nilai | |
|---|---|---|---|---|---|---|---|---|---|
| | | Pengembang | System | TP | FP | FN | TN | Precission | Recall |
| Cerita Pengguna | US-001 | 2 | 2 | 1 | 1 | 1 | 49 | 0.5 | 0.5 |
| | US-002 | 1 | 0 | 0 | 0 | 1 | 51 | 0 | 0 |
| | US-003 | 1 | 1 | 1 | 0 | 0 | 51 | 1 | 1 |
| | US-004 | 4 | 1 | 1 | 0 | 3 | 48 | 1 | 0.25 |
| | US-005 | 5 | 2 | 2 | 0 | 3 | 47 | 1 | 0.4 |
| | US-006 | 4 | 0 | 0 | 0 | 4 | 48 | 0 | 0 |
| | US-007 | 5 | 2 | 1 | 1 | 4 | 46 | 0.5 | 0.2 |
| | US-008 | 2 | 1 | 0 | 1 | 2 | 49 | 0 | 0 |
| | US-009 | 2 | 2 | 1 | 1 | 1 | 49 | 0.5 | 0.5 |
| | US-010 | 2 | 6 | 0 | 6 | 2 | 44 | 0 | 0 |
| | US-011 | 1 | 6 | 0 | 6 | 1 | 45 | 0 | 0 |
| | US-012 | 5 | 1 | 0 | 1 | 5 | 46 | 0 | 0 |
| | US-013 | 3 | 0 | 0 | 0 | 3 | 49 | 0 | 0 |
| | US-014 | 2 | 5 | 1 | 4 | 1 | 46 | 0.2 | 0.5 |
| | US-015 | 2 | 5 | 1 | 4 | 1 | 46 | 0.2 | 0.5 |
| | US-016 | 3 | 1 | 0 | 1 | 3 | 48 | 0 | 0 |
| | US-017 | 3 | 1 | 0 | 1 | 3 | 48 | 0 | 0 |
| | | 47 | 36 | 9 | 27 | 38 | 810 | 0.288235294 | 0.226470588 |
| | | TOTAL | | | | | | Rata-Rata | |

*Keterangan:*

| | |
|---|---|
| TP | True Positive |
| FP | False Positive |
| TN | True Negative |
| FN | False Negative |

Dengan tabel yang sama dilakukan percobaan pada smartPortal dengan masing-masing *similarity threshold* 0.4, 0.5, 0.6, 0.7 dipilih *threshold* tersebut karena semakin kecil *threshold* semakin banyak hasil identifikasi system yang salah, sedangkan semakin tinggi semakin sedikit yang teridentifikasi. Selain *similarity threshold* ditambahkan pula prosentase jumlah kata yang terdapat pada kode sumber dibandingkan dengan jumlah kata yang terdapat pada cerita pengguna dengan *threshold* 0%, 5%, 10%, 15%, 20%, 25%, dan 30%. Dipilih prosentase tersebut dengan alasan yang sama dengan *similarity threshold.*

Berdasarkan pemilihan *threshold* tersebut maka dihasilkan tabel 4.18 Tabel rata-rata precission dan recall smartPortal. Dari tabel ini dibentuk grafik precission, recall serta trend grafik precission dan recall. Pada tabel 4.18 nilai recall tertinggi terdapat pada similarity threshold 0.4 dan 0.5 serta pada kandungan kata 0% dan 5%, hal ini disebabkan terdapat 22 kerunutan yang berhasil teridentifikasi dari 47 yang seharusnya, tetapi system mendeteksi lebih dari 179 kerunutan sehingga nilai

presisinya rendah. Nilai presisi tertinggi terletak pada *similarity threshold* 0.7 dan prosentase kandungan kata 25%, hal ini disebabkan system berhasil mengidentifikasi 9 kerunutan yang tepat dari 47 yang seharusnya ditemukan, dan terdapat 27 kesalahan yang seharusnya bukan merupakan kerunutan.

Tabel 4.18. Rata-rata precission recall *smartPortal*

| No | Similarity Threshold | Kandungan Jml Kata | Jml Identifikasi | | Jumlah Perbandingan | | | | Rata-Rata | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Pengembang | System | TP | FP | FN | TN | Precission | Recall |
| 1 | 0.4 | 0% | 47 | 196 | 22 | 174 | 25 | 663 | 0.115875791 | 0.511764706 |
| 2 | 0.4 | 5% | 47 | 185 | 22 | 163 | 25 | 674 | 0.119867388 | 0.511764706 |
| 3 | 0.4 | 10% | 47 | 140 | 19 | 121 | 28 | 716 | 0.143565911 | 0.448039216 |
| 4 | 0.4 | 15% | 47 | 104 | 17 | 87 | 30 | 750 | 0.16767889 | 0.421568627 |
| 5 | 0.4 | 20% | 47 | 80 | 14 | 66 | 33 | 771 | 0.190896359 | 0.321568627 |
| 6 | 0.4 | 25% | 47 | 56 | 11 | 45 | 36 | 792 | 0.193137255 | 0.257843137 |
| 7 | 0.4 | 30% | 47 | 38 | 9 | 29 | 38 | 808 | 0.251960784 | 0.226470588 |
| 8 | 0.5 | 0% | 47 | 187 | 22 | 165 | 25 | 672 | 0.122930991 | 0.511764706 |
| 9 | 0.5 | 5% | 47 | 179 | 22 | 157 | 25 | 680 | 0.127156014 | 0.511764706 |
| 10 | 0.5 | 10% | 47 | 123 | 18 | 105 | 29 | 732 | 0.163468721 | 0.43627451 |
| 11 | 0.5 | 15% | 47 | 90 | 17 | 73 | 30 | 764 | 0.210784314 | 0.421568627 |
| 12 | 0.5 | 20% | 47 | 67 | 13 | 54 | 34 | 783 | 0.265686275 | 0.301960784 |
| 13 | 0.5 | 25% | 47 | 42 | 10 | 32 | 37 | 805 | 0.26372549 | 0.238235294 |
| 14 | 0.5 | 30% | 47 | 28 | 7 | 21 | 40 | 816 | 0.200980392 | 0.155882353 |
| 15 | 0.6 | 0% | 47 | 174 | 21 | 153 | 26 | 684 | 0.126569509 | 0.497058824 |
| 16 | 0.6 | 5% | 47 | 166 | 21 | 145 | 26 | 692 | 0.130794532 | 0.497058824 |
| 17 | 0.6 | 10% | 47 | 117 | 17 | 100 | 30 | 737 | 0.190919701 | 0.424509804 |
| 18 | 0.6 | 15% | 47 | 79 | 13 | 66 | 34 | 771 | 0.211764706 | 0.301960784 |
| 19 | 0.6 | 20% | 47 | 63 | 12 | 51 | 35 | 786 | 0.255882353 | 0.282352941 |
| 20 | 0.6 | 25% | 47 | 37 | 9 | 28 | 38 | 809 | 0.258823529 | 0.226470588 |
| 21 | 0.6 | 30% | 47 | 24 | 7 | 17 | 40 | 820 | 0.215686275 | 0.155882353 |
| 22 | 0.7 | 0% | 47 | 174 | 21 | 153 | 26 | 684 | 0.126569509 | 0.497058824 |
| 23 | 0.7 | 5% | 47 | 166 | 21 | 145 | 26 | 692 | 0.130794532 | 0.497058824 |
| 24 | 0.7 | 10% | 47 | 115 | 15 | 100 | 32 | 737 | 0.18111578 | 0.33627451 |
| 25 | 0.7 | 15% | 47 | 77 | 12 | 65 | 35 | 772 | 0.201960784 | 0.290196078 |
| 26 | 0.7 | 20% | 47 | 61 | 11 | 50 | 36 | 787 | 0.246078431 | 0.270588235 |
| 27 | 0.7 | 25% | 47 | 36 | 9 | 27 | 38 | 810 | 0.288235294 | 0.226470588 |
| 28 | 0.7 | 30% | 47 | 21 | 7 | 14 | 40 | 823 | 0.245098039 | 0.155882353 |

*Keterangan:*

| | |
|---|---|
| TP | True Positive |
| FP | False Positive |
| TN | True Negative |
| FN | False Negative |

Gambar 4.11 menunjukkan kecenderungan yang sama untuk setiap ambang batas kemiripan kata apabila tidak ditambahkan jumlah kata yg terdapat dalam kode sumber. Nilai precission untuk prosentase jumlah kata 0% hanya berkisar antara

0.116 sampai 0.126. Setelah ditambahkan prosentase jumlah kata yang terkandung, terjadi peningkatan pada nilai presisinya seiring dengan penambahan prosentase jumlah kata dalam kode sumber, puncak nilai tertinggi 0,288 terdapat pada ambang batas 0.7 dan prosentase kata 25%, tetapi mengalami penurunan kembali pada angka 30%.

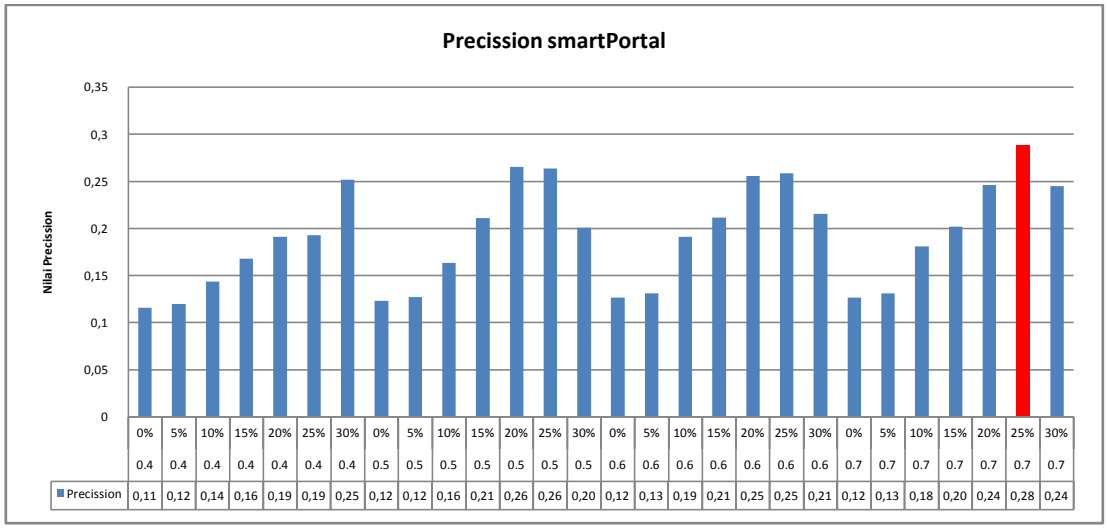
Rendahnya nilai presisi disebabkan oleh identifikasi system yang tidak tepat, ketidaktepatan identifikasi ini akibat dari banyak ditemukannya kata-kata yang sama pada kode sumber tetapi antara kode sumber dan cerita pengguna tidak ada hubungan kerunutan menurut pengembang. Sebagai contoh pada percobaan ke 27, ambang batas kemiripan 0.7 dan besarnya perbandingan jumlah kata dalam kode sumber dibanding cerita pengguna 25%, pada tabel 4.16. Perbandingan identifikasi pengembang dan system dapat dilihat pada cerita pengguna ke 15 (US-015) yang berisi "`Sebagai petugas loket saya harus dapat mencetak hasil penutupan transaksi setiap shift sehingga saya dapat mengetahui total penerimaan selama satu shift setiap petugas`". US-015 menurut pengembang berhubungan dengan kode sumber:

- SC-008 (*/portal-truk2/src/aplikasi/portal/transaksi/PanelTutupTransaksi.java*)
- SC-046 (*/portal-truk2/src/aplikasi/portal/service/ReportService.java*).

Sedangkan menurut system berhubungan dengan 5 kode sumber yaitu:

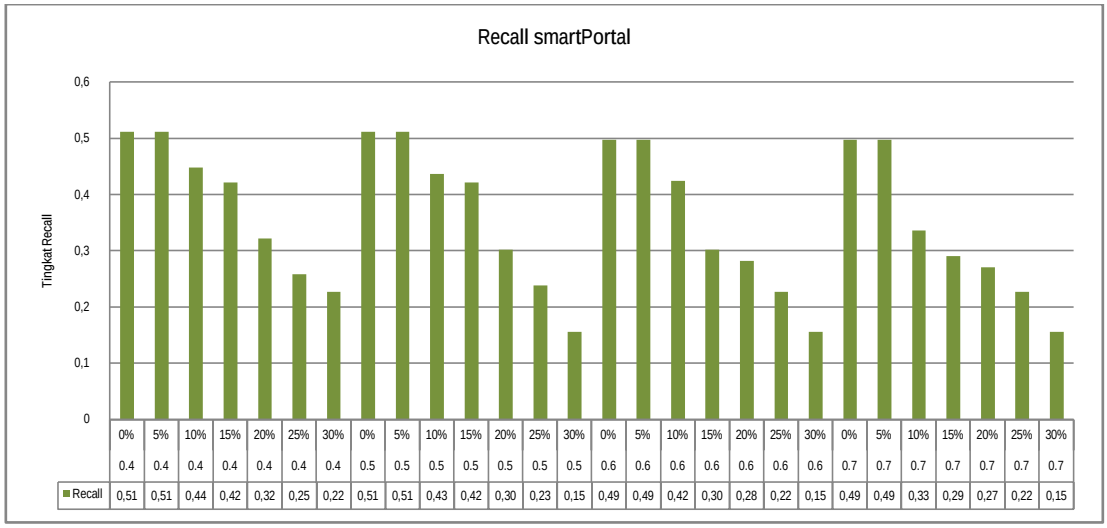
- SC-004 *(/portal-truk2/src/aplikasi/portal/transaksi/FrmTransaksi.java)*
- SC-005 *(/portal-truk2/src/aplikasi/portal/transaksi/DlgImageTrx.java)*
- SC-007 *(/portal-truk2/src/aplikasi/portal/transaksi/PanelTransaksiHistori.java)*
- SC-008 *(/portal-truk2/src/aplikasi/portal/transaksi/PanelTutupTransaksi.java)*
- SC-025 *(/portal-truk2/src/aplikasi/portal/form/FormUtama.java)*

Menurut identifikasi system kode sumber SC-046 tidak teridentifikasi karena hanya ditemukan kata "cetak" yang mirip pada cerita pengguna. Sehingga prosentase jumlah kata tidak memenuhi ambang batas yang ditentukan.

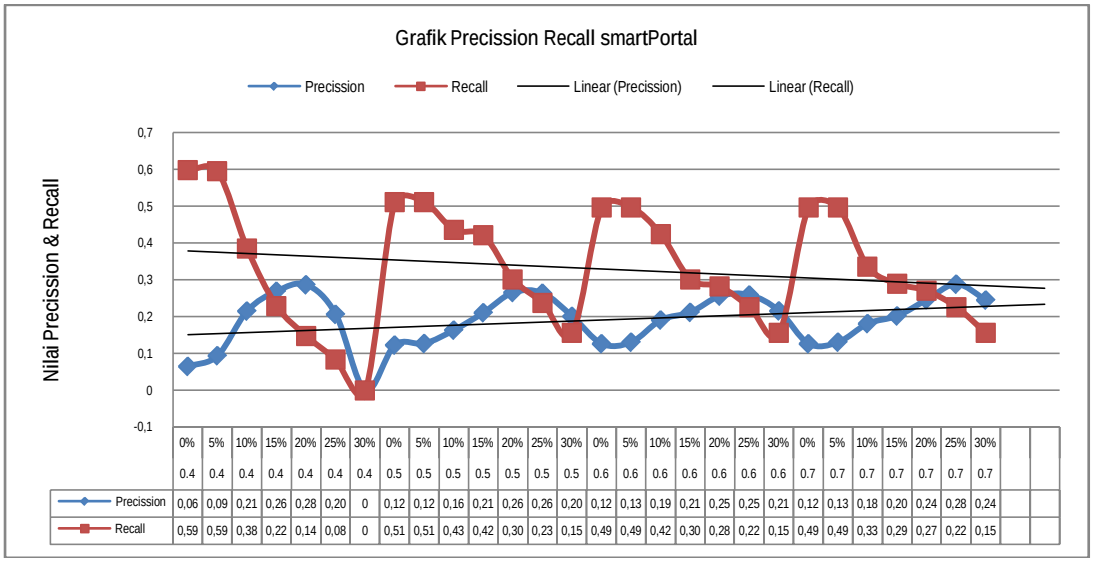



Gambar 4.11. Grafik Precission smartPortal.

Gambar 4.12 menunjukkan kecenderungan yang sama untuk setiap ambang batas kemiripan kata apabila tidak ditambahkan jumlah kata yg terdapat dalam kode sumber. Nilai recall untuk prosentase jumlah kata 0% dan 5% berkisar antara 0.497 sampai 0.512. Setelah ditambahkan prosentase jumlah kata yang terkandung dalam kode sumber, terjadi penurunan pada nilai recall seiring dengan penambahan prosentase jumlah kata dalam kode sumber.



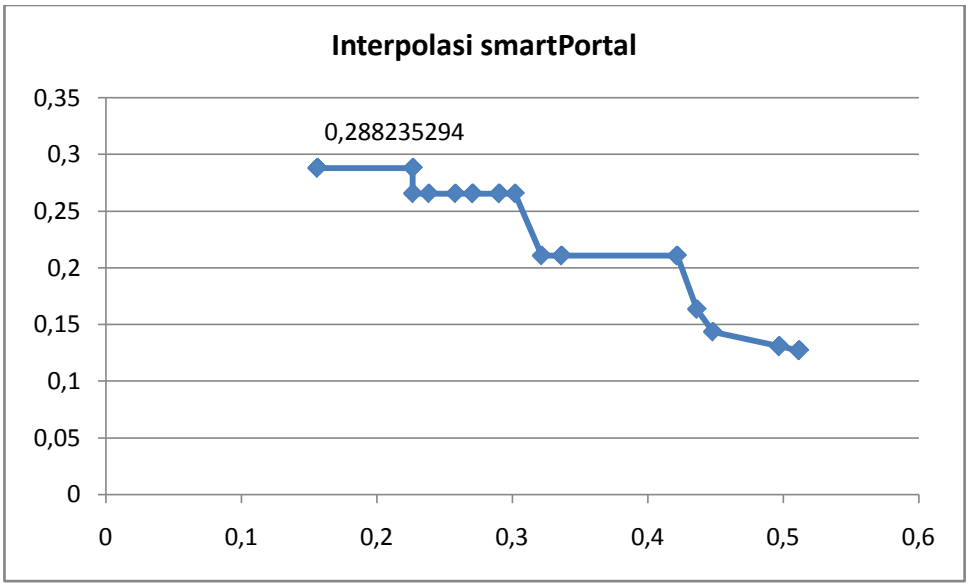

Gambar 4.12. Grafik Recall smartPortal.




Gambar 4.13. Grafik Trend Precission dan Recall smartPortal.

Garis *linier* pada Gambar 4.13 menunjukkan kecenderungan (*trend*) peningkatan nilai presisi yang meningkat seiring dengan peningkatan nilai ambang batas kemiripan, tetapi garis *linier* recall menunjukan kecenderungan menurun seiring dengan peningkatan nilai ambang batas kemiripan.

Berdasarkan gambar 4.14. Kurva interpolasi Precission dan Recall *smartPortal*, Perpotongan antara precission dan recall terletak pada titik antara prosentase jumlah kata 20% dan 25% pada ambang batas kemiripan 0,7 merupakan titik terbaik dari presisi dan recall, dengan nilai presisi antara 0,288 dengan nilai recall antara 0,556 sampai 0,226.




Gambar 4.14. Kurva interpolasi Precission dan Recall smartPortal.

### 4.4.2 Pengujian smartAbsensi

Aplikasi ini digunakan untuk mengintegrasikan data mesin absensi dari berbagai merek dan lokasi yang tersebar diseluruh unit kerja yang berada di salah satu pemerintah kota di Jawa Timur. Terdapat fungsi lain dari aplikasi *smartAbsensi* ini yaitu digunakan untuk memberikan laporan secara langsung melalui *sms-gateway* kepada masing-masing pimpinan unit kerja. Selain itu juga digunakan sebagai media pelaporan masing-masing unit kerja kepada bagian kepegawaian.

*SmartAbsensi* merupakan data uji dengan kriteria sedang yang terdiri dari 47 cerita pengguna dan 161 file kode sumber. Daftar cerita pengguna dapat dilihat pada tabel 4.19. Tabel cerita pengguna *smartAbsensi*. Daftar kode sumber dapat dilihat pada tabel 4.20. Tabel file kode sumber *smartAbsensi*.

Tabel 4.19. Tabel cerita pengguna smartAbsensi

| Kode Cerita Pengguna | Cerita Pengguna |
|---|---|
| US-001 | Sebagai pengguna saya ingin memiliki halaman login sehingga saya dapat masuk kedalam sistem menggunakan user dan password |
| US-002 | Sebagai pengguna saya ingin memiliki tombol logout sehingga saya dapat masuk keluar dari system |
| US-003 | Sebagai superuser saya ingin memiliki halaman dashboard sehingga saya dapat melihat jumlah pegawai, jumlah absensi masuk, jumlah absensi pulang, dan prosentase kehadiran |
| US-004 | Sebagai superuser saya ingin mengelola daftar pegawai sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data pegawai |
| US-005 | Sebagai superuser saya ingin mengelola daftar unit organisasi kerja sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data unit organisasi kerja |
| US-006 | Sebagai superuser saya ingin mengelola daftar status pegawai sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data status pegawai |
| US-007 | Sebagai superuser saya ingin mengelola daftar Golongan sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data Golongan |
| US-008 | Sebagai superuser saya ingin mengelola daftar Eselon sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data Eselon |
| US-009 | Sebagai superuser saya ingin mengelola daftar Jenis Kelamin sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data Jenis Kelamin |
| US-010 | Sebagai superuser saya ingin mengelola daftar Agama sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data Agama |
| US-011 | Sebagai superuser saya ingin mengelola daftar hari libur sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data hari libur |
| US-012 | Sebagai superuser saya ingin mengelola daftar jenis ijin cuti sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data jenis ijin cuti |
| US-013 | Sebagai superuser saya ingin mengelola daftar jenis roster sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data jenis roster |
| US-014 | Sebagai superuser saya ingin mengelola daftar jam kerja sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data jenis roster |
| US-015 | Sebagai admin skpd saya ingin mengelola daftar roster sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data jam kerjai |

| | |
|---|---|
| US-016 | Sebagai admin skpd saya ingin mengelola daftar role jam kerja sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data role jam kerja |
| US-017 | Sebagai superuser saya ingin mengelola daftar mesin absensi sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data mesin absensi |
| US-018 | Sebagai superuser saya ingin mengelola daftar pegawai setiap mesin sehingga saya dapat melakukan mapping data pegawai pada database dengan data pegawai pada mesin |
| US-019 | Sebagai admin skpd saya ingin dapat memasukan ijin cuti pegawai sehingga saya dapat menyimpan ijin cuti setiap pegawai |
| US-020 | Sebagai admin skpd saya ingin dapat mengupload surat ijin cuti pegawai sehingga saya dapat menyimpan surat ijin cuti setiap pegawai |
| US-021 | Sebagai admin skpd saya ingin dapat melihat daftar pengajuan ijin cuti sehingga saya dapat merubah status ijin cuti pegawai |
| US-022 | Sebagai superuser saya ingin melihat daftar log absensi per unit organisi sehingga saya dapat menunjukan kepada pegawai detail absensi mereka |
| US-023 | Sebagai superuser saya ingin melihat daftar log absensi per mesin absensi sehingga saya dapat menunjukan kepada pegawai detail absensi mereka |
| US-024 | Sebagai admin skpd saya ingin melihat daftar log absensi per unit organisi sehingga saya dapat menunjukan kepada pegawai detail absensi mereka |
| US-025 | Sebagai admin skpd saya ingin melihat daftar log absensi per mesin absensi sehingga saya dapat menunjukan kepada pegawai detail absensi mereka |
| US-026 | Sebagai admin skpd saya ingin dapat memasukkan absensi manual sehingga saya dapat tetap memasukkan data absensi jika terjadi kerusakan mesin atau pegawai tidak dapat melakukan absensi melalui mesin |
| US-027 | Sebagai superuser saya ingin dapat mengupload log absensi per mesin absensi sehingga data absensi pada mesin yang tidak terhubung jaringan tetap dapat masuk kedalam system |
| US-028 | Sebagai superuser saya ingin mengelola daftar role user sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data role user |
| US-029 | Sebagai superuser saya ingin mengelola daftar user sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus data user |
| US-030 | Sebagai superuser saya ingin mengelola hak akses atau user permission sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus hak akses atau user permission |
| US-031 | Sebagai superuser saya ingin mengelola hari libur sehingga saya dapat melihat, menambahkan, mengedit, mencari, dan menghapus daftar hari libur |
| US-032 | Sebagai superuser saya ingin menentukan waktu backup database sehingga saya dapat merubah waktu backup database |
| US-033 | Sebagai superuser saya ingin melihat log aktivitas user sehingga saya dapat mengetahui setiap user yang masuk ke system |
| US-034 | Sebagai superuser saya ingin menampilkan laporan absensi per pegawai setiap bulan sehingga saya dapat mengetahui tanggal masuk, jam masuk, jam keluar, selisih keterlambatan, selisih pulang cepat dan jam lembur setiap pegawai |
| US-035 | Sebagai admin skpd saya ingin menampilkan laporan absensi per pegawai setiap bulan sehingga saya dapat mengetahui tanggal masuk, jam masuk, jam keluar, selisih keterlambatan, selisih pulang cepat dan jam lembur setiap pegawai di skpd saya |
| US-036 | Sebagai superuser saya ingin menampilkan rekapitulasi kehadiran per unit kerja pada periode waktu tertentu sehingga saya dapat mengetahui dalam bentuk tabel nip, nama, jabatan, jumlah ketidakhadiran, total hari masuk, total hari keluar, jumlah jam kerja, jumlah datang terlambat, jumlah pulang cepat dan prosentase kehadiran. |
| US-037 | Sebagai admin skpd saya ingin menampilkan rekapitulasi kehadiran per unit kerja pada periode waktu tertentu sehingga saya dapat mengetahui dalam bentuk tabel nip, nama, jabatan, jumlah ketidakhadiran, total hari masuk, total hari keluar, jumlah jam kerja, jumlah datang terlambat, jumlah pulang cepat dan prosentase kehadiran |
| US-038 | Sebagai superuser saya ingin menampilkan rekapitulasi tingkat kehadiran per unit kerja pada bulan tertentu sehingga saya dapat mengetahui dalam bentuk tabel nip, nama, jabatan, jumlah hari kerja, jumlah kehadiran, jumlah sakit, jumlah ijin, jumlah cuti, jumlah dinas luar, jumlah keterlambatan, dan jumlah pulang sebelum waktunya |

| | |
|---|---|
| US-039 | Sebagai admin skpd saya ingin menampilkan rekapitulasi tingkat kehadiran per unit kerja pada bulan tertentu sehingga saya dapat mengetahui dalam bentuk tabel nip, nama, jabatan, jumlah hari kerja, jumlah kehadiran, jumlah sakit, jumlah ijin, jumlah cuti, jumlah dinas luar, jumlah keterlambatan, dan jumlah pulang sebelum waktunya |
| US-040 | Sebagai superuser saya ingin menampilkan rekapitulasi tingkat kehadiran per jenis kelamin pada periode waktu tertentu sehingga saya dapat mengetahui dalam bentuk tabel nip, nama, jabatan, jumlah hari kerja, jumlah kehadiran, jumlah sakit, jumlah ijin, jumlah cuti, jumlah dinas luar, jumlah keterlambatan, dan jumlah pulang cepat |
| US-041 | Sebagai admin skpd saya ingin menampilkan rekapitulasi tingkat kehadiran kehadiran per jenis kelamin pada periode waktu tertentu sehingga saya dapat mengetahui dalam bentuk tabel nip, nama, jabatan, jumlah hari kerja, jumlah kehadiran, jumlah sakit, jumlah ijin, jumlah cuti, jumlah dinas luar, jumlah keterlambatan, dan jumlah pulang sebelum waktunya |
| US-042 | Sebagai superuser saya ingin menampilkan Rekap Absensi Per Unit Kerja pada bulan tertentu sehingga saya dapat mengetahui dalam bentuk tabel nama unit kerja, jumlah jumlah PNS, jumlah NON PNS, jumlah pegawai yang hadir, dinas luar, ijin, cuti, sakit, tugas belajar, masuk tanpa keterangan, dan prosentase kehadiran |
| US-043 | Sebagai superuser saya ingin menampilkan Laporan Detail Waktu pada bulan tertentu per unit organisasi sehingga saya dapat mengetahui dalam bentuk tabel nip, nama pegawai, dan per tanggal |
| US-044 | Sebagai superuser saya ingin menampilkan Laporan Rekap Jam Kerja pada bulan tertentu per unit organisasi sehingga saya dapat mengetahui dalam bentuk tabel nama pegawai, nip, jabatan, jumlah jam kerja per minggu, dan jumlah jam kerja dalam satu bulan |
| US-045 | Sebagai superuser saya ingin menampilkan Laporan Absen terlambat pada periode waktu tertentu per unit organisasi sehingga saya dapat mengetahui dalam bentuk tabel nama pegawai, tanggal, jam absen masuk, jam absen pulang, jadwal masuk, jadwal pulang, selisih datang terlambat, selisih pulang cepat, keterangan, dan rekap jam lebih |
| US-046 | Sebagai superuser saya ingin menampilkan Laporan Kriteria Absensi Per Pegawai pada periode waktu tertentu per pegawai sehingga saya dapat mengetahui dalam bentuk tabel tanggal, jam absen masuk, jam absen pulang, jadwal masuk, jadwal pulang, kriteria absen masuk, kriteria absen pulang, dan keterangan |
| US-047 | Sebagai superuser saya ingin menampilkan Rekap Kriteria Absen per Unit Organisasi Per unit organisasi pada periode waktu tertentu sehingga saya dapat mengetahui dalam bentuk tabel nip, nama pegawai, jumlah hari berdasarkan kriteria keterlambatan, jumlah hari berdasarkan kriteria pulang cepat, dan jumlah hari tidak masuk tanpa keterangan |

File kode sumber seperti tampak pada tabel 4.10 masih belum sepenuhnya menggunakan bahasa Indonesia, hal ini berpengaruh pada hasil ujicoba karena kata-kata yang dihasilkan dari proses *parsing* maupun *stemming* tidak dapat ditemukan atau tidak diperoleh kemiripan katanya pada cerita pengguna.

Tabel 4.20. Tabel kode sumber *smartAbsensi*

| Kode | File Kode Sumber |
|---|---|
| SC-001 | ../absensi/MesinTest.java |
| SC-002 | ../absensi/TestUnorTree.java |
| SC-003 | ../absensi/HashTests.java |
| SC-004 | ../absensi/TestInsertLog.java |
| SC-005 | ../absensi/AppTest.java |
| SC-006 | ../absensi/ReportParam.java |
| SC-007 | ../absensi/crypto/FileEncryption.java |
| SC-008 | ../absensi/crypto/CipherExample.java |
| SC-009 | ../absensi/crypto/CryptoUtilsTest.java |
| SC-010 | ../absensi/crypto/CryptoException.java |
| SC-011 | ../absensi/crypto/CryptoUtils.java |
| SC-012 | ../absensi/web/HariLiburController.java |

| | |
|---|---|
| SC-013 | ../absensi/web/SKPDController.java |
| SC-014 | ../absensi/web/ServiceController.java |
| SC-015 | ../absensi/web/RoleController.java |
| SC-016 | ../absensi/web/MesinUserController.java |
| SC-017 | ../absensi/web/ZkController.java |
| SC-018 | ../absensi/web/AgamaController.java |
| SC-019 | ../absensi/web/OptionsController.java |
| SC-020 | ../absensi/web/JamKerjaController.java |
| SC-021 | ../absensi/web/ApplicationConfigController.java |
| SC-022 | ../absensi/web/PermissionController.java |
| SC-023 | ../absensi/web/TestZkController.java |
| SC-024 | ../absensi/web/UnitOrganisasiKerjaController.java |
| SC-025 | ../absensi/web/EselonController.java |
| SC-026 | ../absensi/web/UserController.java |
| SC-027 | ../absensi/web/LogActivityController.java |
| SC-028 | ../absensi/web/DashboardController.java |
| SC-029 | ../absensi/web/PegawaiController.java |
| SC-030 | ../absensi/web/JenisJabatanController.java |
| SC-031 | ../absensi/web/interceptor/CorsInterceptor.java |
| SC-032 | ../absensi/web/JenisKelaminController.java |
| SC-033 | ../absensi/web/JenisRoosterController.java |
| SC-034 | ../absensi/web/ReportController.java |
| SC-035 | ../absensi/web/MesinController.java |
| SC-036 | ../absensi/web/StatusPegawaiController.java |
| SC-037 | ../absensi/web/JenisCutiController.java |
| SC-038 | ../absensi/web/JenisMesinController.java |
| SC-039 | ../absensi/web/UserPhotoController.java |
| SC-040 | ../absensi/web/IjinCutiPegawaiController.java |
| SC-041 | ../absensi/web/MasterHariLiburController.java |
| SC-042 | ../absensi/web/GolonganController.java |
| SC-043 | ../absensi/web/MenuController.java |
| SC-044 | ../absensi/web/HomepageController.java |
| SC-045 | ../absensi/web/RoleJamKerjaController.java |
| SC-046 | ../absensi/web/AbsensiLogController.java |
| SC-047 | ../absensi/web/RosterController.java |
| SC-048 | ../absensi/App.java |
| SC-049 | ../absensi/Log.java |
| SC-050 | ../absensi/model/ZkUserInfo.java |
| SC-051 | ../absensi/model/UnorTree.java |
| SC-052 | ../absensi/model/AbsenManual.java |
| SC-053 | ../absensi/security/AjaxAwareLoginUrlAuthenticationEntryPoint.java |
| SC-054 | ../absensi/security/RestAuthenticationEntryPoint.java |
| SC-055 | ../absensi/dao/PegawaiRoleJamKerjaHistoryDao.java |
| SC-056 | ../absensi/dao/PermissionDao.java |
| SC-057 | ../absensi/dao/MesinDao.java |
| SC-058 | ../absensi/dao/UnitOrganisasiKerjaDao.java |
| SC-059 | ../absensi/dao/StatusPegawaiDao.java |
| SC-060 | ../absensi/dao/MesinJamDownloadDao.java |
| SC-061 | ../absensi/dao/AbsensiLogDao.java |
| SC-062 | ../absensi/dao/JenisMesinDao.java |

| | |
|---|---|
| SC-063 | ../absensi/dao/UserDao.java |
| SC-064 | ../absensi/dao/JenisKelaminDao.java |
| SC-065 | ../absensi/dao/HariLiburDao.java |
| SC-066 | ../absensi/dao/LogActivityDao.java |
| SC-067 | ../absensi/dao/IjinCutiPegawaiDao.java |
| SC-068 | ../absensi/dao/AbsensiLogUploadDao.java |
| SC-069 | ../absensi/dao/JenisCutiDao.java |
| SC-070 | ../absensi/dao/SKPDDao.java |
| SC-071 | ../absensi/dao/ApplicationConfigDao.java |
| SC-072 | ../absensi/dao/EselonDao.java |
| SC-073 | ../absensi/dao/HomepageDao.java |
| SC-074 | ../absensi/dao/jdbc/RosterJdbcDao.java |
| SC-075 | ../absensi/dao/jdbc/PegawaiDaoJdbc.java |
| SC-076 | ../absensi/dao/jdbc/ReportDao.java |
| SC-077 | ../absensi/dao/jdbc/ReportJdbcDao.java |
| SC-078 | ../absensi/dao/jdbc/IjinCutiPegawaiJdbcDao.java |
| SC-079 | ../absensi/dao/jdbc/ZkTecoWsOld.java |
| SC-080 | ../absensi/dao/jdbc/PgDao.java |
| SC-081 | ../absensi/dao/jdbc/MapQueryAccess.java |
| SC-082 | ../absensi/dao/jdbc/ServiceDao.java |
| SC-083 | ../absensi/dao/jdbc/HariLiburJdbcDao.java |
| SC-084 | ../absensi/dao/jdbc/DashboardJdbc.java |
| SC-085 | ../absensi/dao/jdbc/UserInfoNitgenDao.java |
| SC-086 | ../absensi/dao/jdbc/ZkTecoWs.java |
| SC-087 | ../absensi/dao/jdbc/MapResultSet.java |
| SC-088 | ../absensi/dao/jdbc/AbsensiLogDaoJdbc.java |
| SC-089 | ../absensi/dao/jdbc/UserInfoZkDao.java |
| SC-090 | ../absensi/dao/MenuDao.java |
| SC-091 | ../absensi/dao/JamKerjaDao.java |
| SC-092 | ../absensi/dao/RoleDao.java |
| SC-093 | ../absensi/dao/JenisJabatanDao.java |
| SC-094 | ../absensi/dao/RoleJamKerjaDetailDao.java |
| SC-095 | ../absensi/dao/PegawaiRoleHistorysDao.java |
| SC-096 | ../absensi/dao/MesinUserDao.java |
| SC-097 | ../absensi/dao/MasterHariLiburDao.java |
| SC-098 | ../absensi/dao/PegawaiDao.java |
| SC-099 | ../absensi/dao/JenisRosterDao.java |
| SC-100 | ../absensi/dao/AgamaDao.java |
| SC-101 | ../absensi/dao/ZkUserInfoDao.java |
| SC-102 | ../absensi/dao/GolonganDao.java |
| SC-103 | ../absensi/dao/RoleJamKerjaDao.java |
| SC-104 | ../absensi/dao/RosterDao.java |
| SC-105 | ../absensi/RosterView.java |
| SC-106 | ../absensi/domain/ApplicationConfig.java |
| SC-107 | ../absensi/domain/IjinCutiStatus.java |
| SC-108 | ../absensi/domain/JenisKelamin.java |
| SC-109 | ../absensi/domain/JenisJabatan.java |
| SC-110 | ../absensi/domain/Mesin.java |
| SC-111 | ../absensi/domain/Menu.java |
| SC-112 | ../absensi/domain/PegawaiRoleJamKerjaHistori.java |

| | |
|---|---|
| SC-113 | ../absensi/domain/RoleJamKerja.java |
| SC-114 | ../absensi/domain/Permission.java |
| SC-115 | ../absensi/domain/StatusAbsen.java |
| SC-116 | ../absensi/domain/Role.java |
| SC-117 | ../absensi/domain/ZkUserInfo.java |
| SC-118 | ../absensi/domain/Hari.java |
| SC-119 | ../absensi/domain/Pegawai.java |
| SC-120 | ../absensi/domain/JamKerja.java |
| SC-121 | ../absensi/domain/MesinUserTemplate.java |
| SC-122 | ../absensi/domain/BaseEntity.java |
| SC-123 | ../absensi/domain/Agama.java |
| SC-124 | ../absensi/domain/User.java |
| SC-125 | ../absensi/domain/LogActivity.java |
| SC-126 | ../absensi/domain/MesinJamDownload.java |
| SC-127 | ../absensi/domain/HariLiburMaster.java |
| SC-128 | ../absensi/domain/MesinUser.java |
| SC-129 | ../absensi/domain/IjinCutiPegawaiDetail.java |
| SC-130 | ../absensi/domain/Golongan.java |
| SC-131 | ../absensi/domain/IjinCutiPegawai.java |
| SC-132 | ../absensi/domain/RoleJamKerjaDetail.java |
| SC-133 | ../absensi/domain/UnitOrganisasiKerja.java |
| SC-134 | ../absensi/domain/SKPD.java |
| SC-135 | ../absensi/domain/StatusPegawai.java |
| SC-136 | ../absensi/domain/UserInfo.java |
| SC-137 | ../absensi/domain/JenisMesin.java |
| SC-138 | ../absensi/domain/PegawaiUnorHistori.java |
| SC-139 | ../absensi/domain/DownloadLog.java |
| SC-140 | ../absensi/domain/ResultSetFunction.java |
| SC-141 | ../absensi/domain/Roster.java |
| SC-142 | ../absensi/domain/AbsensiLogUpload.java |
| SC-143 | ../absensi/domain/HariLibur.java |
| SC-144 | ../absensi/domain/Eselon.java |
| SC-145 | ../absensi/domain/JenisIjinCuti.java |
| SC-146 | ../absensi/domain/AbsensiLog.java |
| SC-147 | ../absensi/domain/JenisRoster.java |
| SC-148 | ../absensi/domain/RowInfo.java |
| SC-149 | ../absensi/service/ReportService.java |
| SC-150 | ../absensi/service/MonitoredService.java |
| SC-151 | ../absensi/service/AbsenService.java |
| SC-152 | ../absensi/service/impl/AbsenServiceImpl.java |
| SC-153 | ../absensi/service/impl/ReportServiceImpl.java |
| SC-154 | ../absensi/service/impl/ConfigServiceImpl.java |
| SC-155 | ../absensi/service/ConfigService.java |
| SC-156 | ../absensi/hash/HashCode.java |
| SC-157 | ../absensi/zkteco/IZKEM.java |
| SC-158 | ../absensi/zkteco/ZkTecoWsNew.java |
| SC-159 | ../absensi/zkteco/MesinZk.java |
| SC-160 | ../absensi/zkteco/_IZKEMEvents.java |
| SC-161 | ../absensi/zkteco/ClassFactory.java |

Berdasarkan tabel cerita pengguna dan tabel kode sumber, pengembang malakukan identifikasi identifikasi hubungan kerunutan sehingga diidentifikasi terdapat 200 hubungan kerunutan, seperti terlihat pada tabel 4.21. Tabel rata-rata precission recall *smartAbsensi* pada kolom ketiga.

Pada tabel tersebut juga dapat dilihat semakin kecil *similarity threshold* semakin banyak kerunutan yang teridentifikasi oleh system pada ambang batas 0.4 dan prosentase kata 0% system mengidentifikasi sebanyak 2092 kerunutan, dan yang benar sesuai dengan identifikasi pengembang hanya 129 dan yang 1963 sistem salah mengidentifikasi sehingga nilai presisi menjadi kecil. Dengan menambahkan prosentase jumlah kata yang terkandung dalam kode program dapat memperbaiki nilai presisi pada percobaan dapat dilihat pada tabel 4.21 dengan menambahkan prosentase antara 15% sampai 20% nilai presisi menjadi lebih baik.

Hampir sama dengan smartPortal, pemberian jumlah kandungan kata diatas 20% akan mengurangi jumlah kerunutan yang teridentifikasi, sehingga nilai presisipun menurun. Pada smartAbsensi terdapat sedikit perbedaan karakteristik data yaitu nilai presisi mulai menurun pada saat prosentase jumlah kata sebesar 25% dan mengalami puncaknya pada prosentase 20% dengan nilai presisi sebesar 0.28, dapat dilihat pada gambar 4.15. Grafik precision smartAbsensi.

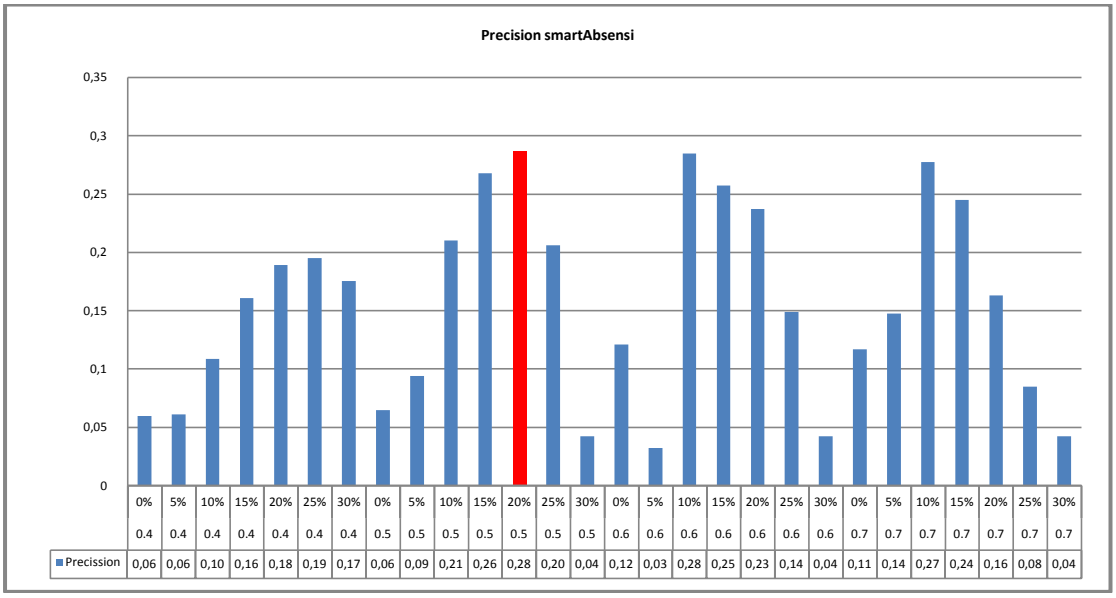
Tabel 4.21. Hasil evaluasi precision recall *smartAbsensi*

| No | Similarity Threshold | Kandungan Jml Kata | Jml Identifikasi | | Jumlah Perbandingan | | | | Rata-Rata | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Pengembang | System | TP | FP | FN | TN | Precission | Recall |
| 1 | 0.4 | 0% | 200 | 2092 | 129 | 1963 | 71 | 5404 | 0.064404687 | 0.605319149 |
| 2 | 0.4 | 5% | 200 | 1583 | 128 | 1455 | 72 | 5912 | 0.089496289 | 0.60177305 |
| 3 | 0.4 | 10% | 200 | 424 | 79 | 345 | 121 | 7022 | 0.215223872 | 0.385815603 |
| 4 | 0.4 | 15% | 200 | 198 | 50 | 148 | 150 | 7219 | 0.268935261 | 0.229432624 |
| 5 | 0.4 | 20% | 200 | 92 | 31 | 61 | 169 | 7306 | 0.286609254 | 0.14822695 |
| 6 | 0.4 | 25% | 200 | 45 | 18 | 27 | 182 | 7340 | 0.206382979 | 0.083333333 |
| 7 | 0.4 | 30% | 200 | 23 | 4 | 19 | 196 | 7348 | 0.063829787 | 0.018439716 |
| 8 | 0.5 | 0% | 200 | 2036 | 128 | 1908 | 72 | 5459 | 0.065040455 | 0.59822695 |
| 9 | 0.5 | 5% | 200 | 1508 | 127 | 1381 | 73 | 5986 | 0.094046193 | 0.594680851 |
| 10 | 0.5 | 10% | 200 | 380 | 74 | 306 | 126 | 7061 | 0.210638564 | 0.364539007 |
| 11 | 0.5 | 15% | 200 | 187 | 49 | 138 | 151 | 7229 | 0.267975582 | 0.222340426 |
| 12 | 0.5 | 20% | 200 | 91 | 31 | 60 | 169 | 7307 | 0.286609254 | 0.14822695 |
| 13 | 0.5 | 25% | 200 | 43 | 17 | 26 | 183 | 7341 | 0.206382979 | 0.078014184 |
| 14 | 0.5 | 30% | 200 | 20 | 2 | 18 | 198 | 7349 | 0.042553191 | 0.008865248 |
| 15 | 0.6 | 0% | 200 | 1436 | 128 | 1308 | 72 | 6059 | 0.121149654 | 0.59822695 |
| 16 | 0.6 | 5% | 200 | 1184 | 39 | 1145 | 161 | 6222 | 0.032703859 | 0.182742317 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 17 | 0.6 | 10% | 200 | 243 | 70 | 173 | 130 | 7194 | 0.285218328 | 0.336170213 |
| 18 | 0.6 | 15% | 200 | 101 | 39 | 62 | 161 | 7305 | 0.257801418 | 0.164184397 |
| 19 | 0.6 | 20% | 200 | 44 | 21 | 23 | 179 | 7344 | 0.237588652 | 0.090070922 |
| 20 | 0.6 | 25% | 200 | 18 | 9 | 9 | 191 | 7358 | 0.14893617 | 0.035815603 |
| 21 | 0.6 | 30% | 200 | 8 | 2 | 6 | 198 | 7361 | 0.042553191 | 0.008865248 |
| 22 | 0.7 | 0% | 200 | 1430 | 124 | 1306 | 76 | 6061 | 0.1169801 | 0.576950355 |
| 23 | 0.7 | 5% | 200 | 1042 | 123 | 919 | 77 | 6448 | 0.147923853 | 0.573404255 |
| 24 | 0.7 | 10% | 200 | 235 | 65 | 170 | 135 | 7197 | 0.277550464 | 0.318439716 |
| 25 | 0.7 | 15% | 200 | 93 | 34 | 59 | 166 | 7308 | 0.245390071 | 0.142907801 |
| 26 | 0.7 | 20% | 200 | 37 | 17 | 20 | 183 | 7347 | 0.163120567 | 0.072340426 |
| 27 | 0.7 | 25% | 200 | 12 | 6 | 6 | 194 | 7361 | 0.085106383 | 0.025177305 |
| 28 | 0.7 | 30% | 200 | 5 | 2 | 3 | 198 | 7364 | 0.042553191 | 0.008865248 |

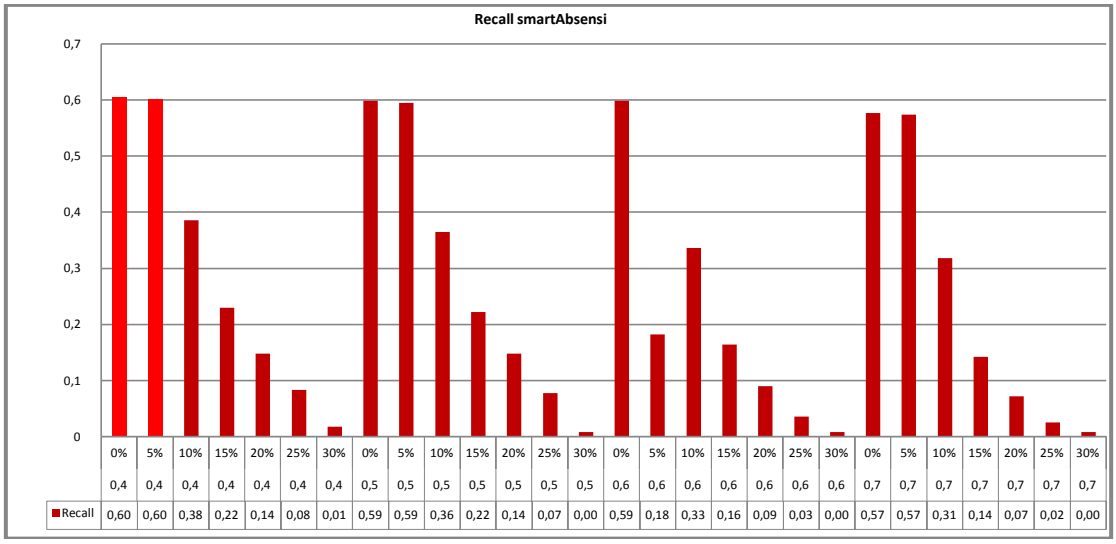
*Keterangan:*

| | |
|---|---|
| TP | True Positive |
| FP | False Positive |
| TN | True Negative |
| FN | False Negative |

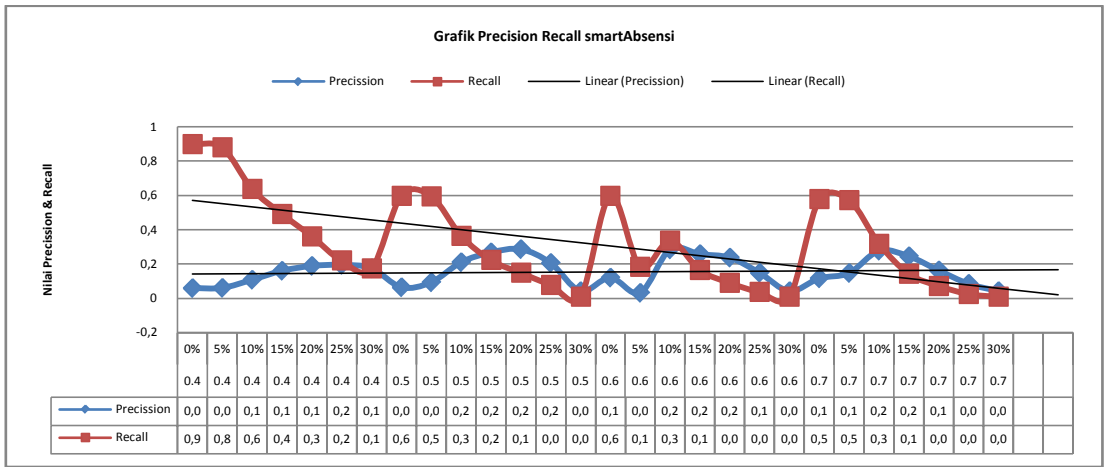



Gambar 4.15. Grafik precision smartAbsensi.

Gambar 4.15 menunjukkan kecenderungan yang sama untuk setiap ambang batas kemiripan kata apabila tidak ditambahkan jumlah kata yg terdapat dalam kode sumber. Nilai recall untuk prosentase jumlah kata 0% dan 5% berkisar antara 0.59 sampai 0.6. Setelah ditambahkan prosentase jumlah kata yang terkandung dalam kode sumber, terjadi penurunan pada nilai recall seiring dengan penambahan prosentase jumlah kata dalam kode sumber.



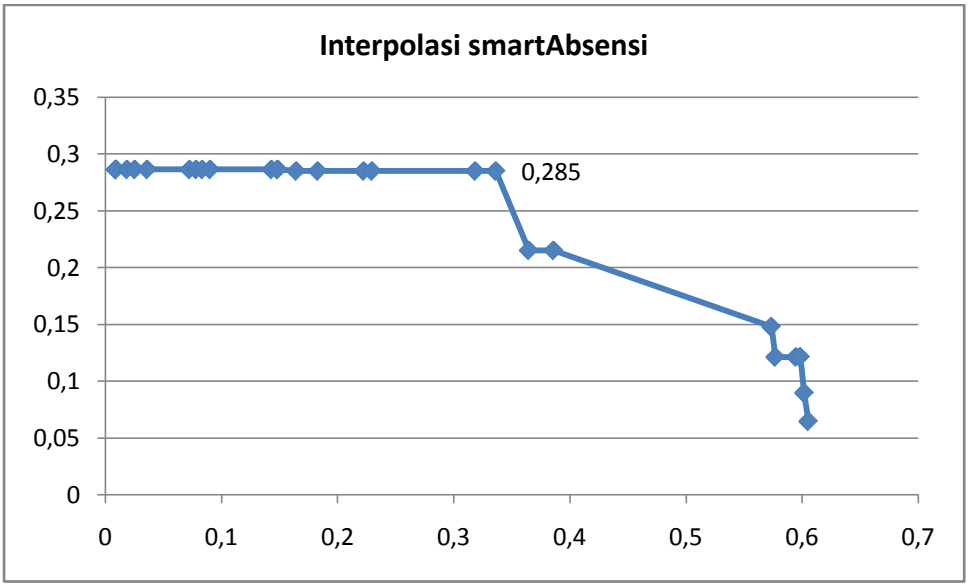

Gambar 4.16. Grafik Recall smartAbsensi




Gambar 4.17. Grafik Trend smartAbsensi

Garis *linier* pada Gambar 4.17 menunjukkan kecenderungan (*trend*) peningkatan nilai presisi yang meningkat seiring dengan peningkatan nilai ambang batas kemiripan, tetapi garis *linier* recall menunjukan kecenderungan menurun seiring dengan peningkatan nilai ambang batas kemiripan.

Berdasarkan gambar 4.18. Kurva interpolasi Precission dan Recall *smartAbsensi*, Perpotongan antara precission dan recall terletak pada titik prosentase jumlah kata 10% pada ambang batas kemiripan 0,6 merupakan titik terbaik dari presisi dan recall, dengan nilai 0,285 berada pada nilai recall 0,008 sampai 0,336.




Gambar 4.18. Grafik Interpolasi smartAbsensi

### 4.4.3 Pengujian smartTravel

Aplikasi ini digunakan untuk melakukan resevasi dan pembelian tiket pesawat, paket tour, hotel, dan penyewaan mobil. Selain fungsi transaksional tersebut, aplikasi ini juga digunakan untuk membantu manajemen dalam membuat laporan kepada *holding company*. Fungsi utama dari aplikasi ini adalah dapat secara otomatis melakukan booking tiket dan mengambil data dari maskapai dengan teknik *web crawling*.

SmartTravel dibangun dengan arsitektur MVC, dimana program java hanya digunakan sebagai penyedia service (*web service)* menggunakan *spring framework*. Sedangkan untuk *frontend* menggunakan HTML5 dengan *angularJS framework*.

*SmartTravel* merupakan data uji dengan kriteria besar yang terdiri dari 80 cerita pengguna dan 222 file kode sumber. Daftar cerita pengguna dapat dilihat pada tabel 4.22. Tabel cerita pengguna *smartTravel*. Daftar kode sumber dapat dilihat pada tabel 4.23. Tabel file kode sumber *smartTravel*.

Tabel 4.22. Rata-rata precission recall *smartAbsensi*

| Kode US | User Story |
|---|---|
| US-001 | Sebagai pengguna saya ingin memiliki halaman login sehingga saya dapat masuk kedalam sistem menggunakan user dan password |
| US-002 | Sebagai pengguna saya ingin memiliki tombol logout sehingga saya dapat masuk keluar dari sistem |
| US-003 | Sebagai manager saya harus dapat memasukkan budget per cabang per bulan per tahun sehingga saya dapat menyimpan budget didalam sistem |
| US-004 | Sebagai manager saya harus dapat melihat target atau budget dengan realisasi penjualan per cabang per bulan per tahun |

| | |
|---|---|
| | sehingga saya dapat menyimpan budget didalam sistem |
| US-005 | Sebagai admin saya harus dapat mengelola master anggaran sehingga saya dapat menambah, merubah, menghapus, dan mencari anggaran |
| US-006 | Sebagai admin saya harus dapat mengelola master relasi sehingga saya dapat menambah, merubah, menghapus, dan mencari relasi |
| US-007 | Sebagai admin saya harus dapat mengelola master pekerja sehingga saya dapat menambah, merubah, menghapus, dan mencari pekerja |
| US-008 | Sebagai admin saya harus dapat mengelola master kota sehingga saya dapat menambah, merubah, menghapus, dan mencari kota |
| US-009 | Sebagai admin saya harus dapat mengelola master provinsi sehingga saya dapat menambah, merubah, menghapus, dan mencari provinsi |
| US-010 | Sebagai admin saya harus dapat mengelola master negara sehingga saya dapat menambah, merubah, menghapus, dan mencari negara |
| US-011 | Sebagai admin saya harus dapat mengelola master Cabang sehingga saya dapat menambah, merubah, menghapus, dan mencari Cabang |
| US-012 | Sebagai admin saya harus dapat mengelola master Item sehingga saya dapat menambah, merubah, menghapus, dan mencari Item |
| US-013 | Sebagai admin saya harus dapat mengelola master Satuan sehingga saya dapat menambah, merubah, menghapus, dan mencari Satuan |
| US-014 | Sebagai admin saya harus dapat mengelola master Akun sehingga saya dapat menambah, merubah, menghapus, dan mencari Akun |
| US-015 | Sebagai admin saya ingin dapat melihat daftar user sehingga saya dapat menambah, merubah, menghapus, dan mencari user |
| US-016 | Sebagai admin saya harus dapat melihat daftar role sehingga saya dapat menambah, merubah, menghapus, dan mencari role |
| US-017 | Sebagai admin saya harus dapat mengelola hak akses menu setiap role sehingga saya dapat membatasi hak akses usur berdasarkan rolenya |
| US-018 | Sebagai admin saya harus dapat melihat daftar menu sehingga saya dapat menambah, merubah, menghapus, dan mencari menu |
| US-019 | Sebagai admin saya harus dapat melihat daftar permission sehingga saya dapat menentukan hak akses setiap user |
| US-020 | Sebagai sales saya ingin dapat melakukan ticket issued sehingga saya dapat melakukannya dimana saja |
| US-021 | Sebagai sales saya ingin dapat melakukan ticket issued secara otomatis dengan memasukkan kode booking sehingga saya tidak perlu memasukan ulang data penumpang |
| US-022 | Sebagai sales saya ingin dapat melakukan ticket issued secara otomatis dengan memasukkan kode booking sehingga saya tidak perlu memasukan ulang rute penerbangan |
| US-023 | Sebagai sales saya ingin dapat melakukan ticket issued secara otomatis dengan memasukkan kode booking sehingga saya tidak perlu memasukan ulang detail tarif |
| US-024 | Sebagai sales saya ingin dapat membuat faktur tiket pesawat berdasarkan nama perusahaan pelanggan sehingga saya dapat memilih tiket-tiket penumpang mana yang akan dicetak |
| US-025 | Sebagai sales saya ingin dapat membuat faktur tiket non pesawat berdasarkan nama perusahaan pelanggan sehingga saya dapat menambahkan tiket non pesawat yang akan dicetak |
| US-026 | Sebagai sales saya ingin dapat membuat faktur tour dan travel berdasarkan nama perusahaan pelanggan sehingga saya dapat menambahkan tiket tour dan travel yang akan dicetak |
| US-027 | Sebagai sales saya ingin dapat membuat faktur voucher hotel berdasarkan nama perusahaan pelanggan sehingga saya dapat menambahkan voucher hotel yang akan dicetak |
| US-028 | Sebagai sales saya ingin dapat membuat faktur Pengurusan Paspor berdasarkan nama perusahaan pelanggan sehingga saya dapat menambahkan Pengurusan Paspor yang akan dicetak |
| US-029 | Sebagai sales saya ingin dapat membatalkan atau void Tiket Pesawat sehingga saya dapat membatalkan tiket yang salah dibuat |
| US-030 | Sebagai kasir saya harus dapat memasukkan pembayaran faktur sehingga saya dapat menyimpan data pembayaran faktur pada sistem |
| US-031 | Sebagai kasir saya harus dapat membuat tagihan PTK sehingga saya dapat menyediakan permintaan tagihan dari PT PETROKIMIA |
| US-032 | Sebagai kasir saya harus dapat membuat tagihan standar atau TTR sehingga saya dapat menyediakan permintaan tagihan standar atau TTR dari pelanggan selain PT PETROKIMIA |
| US-033 | Sebagai kasir saya harus dapat merubah jenis transaksi faktur sehingga saya dapat merubah jenis transaksi kredit ke tunai atau sebaliknya apabila terjadi kesalahan |
| US-034 | Sebagai kasir saya harus dapat mencari faktur yang telah dibuat sehingga saya dapat mengecek faktur apakah faktur tersebut sudah dibuatkan tagihan atau belum |
| US-035 | Sebagai kasir saya harus dapat mencetak rekap Penjualan Petrokimia sehingga saya dapat mencetak Laporan Berdasarkan Kode Anggaran dan Laporan Berdasarkan Rute |
| US-036 | Sebagai kasir saya harus dapat mencetak rekap penjualan Bulanan sehingga saya dapat mengetahui Laporan penjualan per bulan |
| US-037 | Sebagai kasir saya harus dapat mencetak Rincian Faktur - Anggaran sehingga saya dapat mengetahui Laporan rincian faktur per anggaran |
| US-038 | Sebagai kasir saya harus dapat memonitor Faktur sehingga saya dapat mengetahui status faktur tunai atau kredit |
| US-039 | Sebagai kasir saya harus dapat merekap Tagihan PTK sehingga saya dapat mengetahui rincian tagihan per maskapai, nama, rute, kelas, dan harga |
| US-040 | Sebagai bagian boderel saya harus dapat melakukan import boderel per maskapai sehingga saya dapat tidak perlu memasukkan boderel secara manual |
| US-041 | Sebagai bagian boderel saya harus dapat melakukan monitoring boderel sehingga saya dapat mengetahui tiket yang telah dibuatkan faktur atau belum |
| US-042 | Sebagai akunting saya harus dapat memasukkan jurnal manual sehingga saya tetap dapat menjurnal diluar jurnal otomatis dari sistem |

| | |
|---|---|
| US-043 | Sebagai akunting saya harus dapat membuat pengajuan kas bank masuk sehingga saya dapat melakukan pengajuan penerimaan uang selain dari penjualan barang atau jasa |
| US-044 | Sebagai akunting saya harus dapat membuat pengajuan kas bank keluar sehingga saya dapat melakukan entri pengajuan pengeluaran kas atau bank |
| US-045 | Sebagai akunting saya harus dapat membuat kas bank masuk sehingga saya dapat melakukan penerimaan uang selain dari penjualan barang atau jasa |
| US-046 | Sebagai akunting saya harus dapat membuat kas bank keluar sehingga saya dapat melakukan entri pengeluaran kas atau bank |
| US-047 | Sebagai kasir saya ingin melihat history daftar Outstanding tiket Tunai sehingga saya dapat melihat no tiket dan nama perusahaan yg belum terbayar |
| US-048 | Sebagai kasir saya ingin melihat history Faktur Outstanding Tunai sehingga saya dapat melihat faktur berdasarkan no tiket dan nama perusahaan yg belum terbayar |
| US-049 | Sebagai kasir saya ingin melihat history Faktur sehingga saya dapat melihat faktur beserta statusnya |
| US-050 | Sebagai kasir saya ingin melihat Histori Pengajuan Bukti Kas atau Bank sehingga saya dapat melihat detail Pengajuan Kas atau Bank |
| US-051 | Sebagai kasir saya ingin melihat Histori Bukti Kas atau Bank sehingga saya dapat melihat, mencetak, mengedit, dan menghapus bukti Kas atau Bank |
| US-052 | Sebagai kasir saya ingin melihat Histori Tiket Pesawat sehingga saya dapat melihat dan mencari tiket pesawat berdasarkan nama, perusahaan dan periode waktu tertentu |
| US-053 | Sebagai kasir saya ingin melihat Histori Tagihan PTK per periode waktu tertentu sehingga saya dapat mencari, merubah, menghapus, mencetak dalam bentuk pdf, excel atau doc |
| US-054 | Sebagai manajer saya ingin melihat History Approve Koreksi Faktur per periode waktu tertentu sehingga saya dapat mencari, mengetahui status faktur dan menyetujui perubahan faktur |
| US-055 | Sebagai manajer saya ingin melihat History Laporan Penerimaan Penjualan - LPP per periode waktu tertentu sehingga saya dapat dapat menyetujui, mencari, merubah, menghapus, mencetak dalam bentuk pdf, excel atau doc |
| US-056 | Sebagai manajer saya ingin melihat History Pembayaran per cabang per periode waktu tertentu sehingga saya dapat melihat detail, mencetak dan merubahnya |
| US-057 | Sebagai manajer saya ingin melihat log Penerimaan Penjualan - LPP per periode waktu tertentu sehingga saya dapat melihat log dari pembuatan Laporan Penerimaan Penjualan - LPP |
| US-058 | Sebagai manajer saya ingin melihat log tagihan per periode waktu tertentu sehingga saya dapat melihat log dari pembuatan tagihan |
| US-059 | Sebagai manajer saya ingin mencari faktur tiket pesawat per cabang per periode waktu tertentu sehingga saya dapat mengetahui status pembatalan fakturnya |
| US-060 | Sebagai manajer saya ingin melihat log pembayaran per periode waktu tertentu sehingga saya dapat melihat detail pembayaran |
| US-061 | Sebagai bagian boderel saya ingin melihat Histori Boderel setiap maskapai sehingga saya dapat melihat detail setiap tiket |
| US-062 | Sebagai akunting saya ingin melihat Histori Jurnal per kelompok jurnal, per cabang atau per periode waktu tertentu sehingga saya bisa melihat detail per jurnal |
| US-063 | Sebagai akunting saya ingin melihat Histori Jurnal per kelompok jurnal, per cabang atau per periode waktu tertentu sehingga saya bisa melihat detail per jurnal |
| US-064 | Sebagai manajer saya ingin melihat laporan penjualan tiket pesawat sehingga saya dapat melihatnya dalam bentuk pdf maupun xls per cabang, per periode waktu, per jenis pembayaran |
| US-065 | Sebagai manajer saya ingin melihat laporan penjualan tiket non pesawat sehingga saya dapat melihatnya dalam bentuk pdf maupun xls per cabang, per periode waktu, per jenis pembayaran |
| US-066 | Sebagai manajer saya ingin melihat laporan penjualan tour dan travel sehingga saya dapat melihatnya dalam bentuk pdf maupun xls per cabang, per periode waktu, per jenis pembayaran |
| US-067 | Sebagai manajer saya ingin melihat laporan penjualan voucher hotel sehingga saya dapat melihatnya dalam bentuk pdf maupun xls per cabang, per periode waktu, per jenis pembayaran |
| US-068 | Sebagai manajer saya ingin melihat laporan penjualan sewa kendaraan sehingga saya dapat melihatnya dalam bentuk pdf maupun xls per cabang, per periode waktu, per jenis pembayaran |
| US-069 | Sebagai manajer saya ingin melihat laporan penjualan pengurusan pasport sehingga saya dapat melihatnya dalam bentuk pdf maupun xls per cabang, per periode waktu, per jenis pembayaran |
| US-070 | Sebagai manajer saya ingin melihat laporan penjualan ongkos naik haji sehingga saya dapat melihatnya dalam bentuk pdf maupun xls per cabang, per periode waktu, per jenis pembayaran |
| US-071 | Sebagai manajer saya ingin melihat laporan void tiket sehingga saya dapat melihatnya dalam bentuk pdf maupun xls per cabang, per periode waktu, per jenis pembayaran |
| US-072 | Sebagai manajer saya ingin melihat laporan penjualan tiket harian sehingga saya dapat melihatnya dalam bentuk pdf maupun xls per cabang, per periode waktu, per jenis pembayaran |
| US-073 | Sebagai manajer saya ingin melihat laporan penjualan unit travel harian sehingga saya dapat melihatnya dalam bentuk pdf maupun xls per cabang, per periode waktu, per jenis pembayaran |
| US-074 | Sebagai akunting saya ingin melihat Laporan Buku Besar per cabang per periode waktu tertentu per kode rekening sehingga saya dapat melihatnya dalam bentuk pdf maupun xls |
| US-075 | Sebagai akunting saya ingin melihat Laporan Neraca Lajur per cabang per periode waktu tertentu per kode rekening sehingga saya dapat melihat neraca awal, mutasi dan neraca akhir dalam bentuk pdf maupun xls |
| US-076 | Sebagai akunting saya ingin melihat Laporan rugi laba per cabang per bulan tertentu sehingga saya dapat melihatnya dalam bentuk pdf maupun xls |
| US-077 | Sebagai akunting saya ingin melihat Laporan neraca per cabang per bulan tertentu sehingga saya dapat melihatnya saldo neraca bulan lalu dan bulan ini dalam bentuk pdf maupun xls |
| US-078 | Sebagai akunting saya ingin melihat Laporan Aging atau umur Piutang sehingga saya dapat melihat umur piutang pada jangka waktu tertentu per pelanggan dalam bentuk pdf maupun xls |
| US-079 | Sebagai akunting saya ingin melihat Laporan Penerimaan Penjualan- LPP per tanggal tertentu sehingga saya dapat menambahkan data pembayaran per pelanggan |

| | |
|---|---|
| US-080 | Sebagai manajer saya ingin melihat laporan Realisasi Kuantum Penjualan - RKP sehingga saya dapat melihat Realisasi Kuantum Penjualan per bulan per maskapai dalam bentuk pdf maupun xls |

File kode sumber seperti tampak pada tabel 4.23 masih sama dengan dataset sebelumnya, belum sepenuhnya menggunakan bahasa Indonesia, hal ini berpengaruh pada hasil ujicoba karena kata-kata yang dihasilkan dari proses *parsing* maupun *stemming* tidak dapat ditemukan atau tidak diperoleh kemiripan katanya pada cerita pengguna, selain itu dalam file kode sumber ini masih ditemukan file-file program yang tidak digunakan.

Data yang diberikan oleh pengembang tentang matrik kerunutan hubungan untuk aplikasi ini menunjukkan beberapa file kode sumber belum terhubung dengan cerita pengguna.

Tabel 4.23. file Kode Sumber *smartTravel*

| Kode SC | Source Code |
|---|---|
| SC-001 | ../src/main/java/com/gw/greatsoft/booking/CobaBacaAirAsia.java |
| SC-002 | ../src/main/java/com/gw/greatsoft/booking/CobaBacaCitilink.java |
| SC-003 | ../src/main/java/com/gw/greatsoft/booking/Lion.java |
| SC-004 | ../src/main/java/com/gw/greatsoft/booking/test.java |
| SC-005 | ../src/main/java/com/gw/greatsoft/booking/CobaBacaLion.java |
| SC-006 | ../src/main/java/com/gw/greatsoft/booking/Citilink1.java |
| SC-007 | ../src/main/java/com/gw/greatsoft/booking/CobaBacaSriwijaya.java |
| SC-008 | ../src/main/java/com/gw/greatsoft/booking/Sriwijaya.java |
| SC-009 | ../src/main/java/com/gw/greatsoft/booking/GarudaText.java |
| SC-010 | ../src/main/java/com/gw/greatsoft/booking/Citilink.java |
| SC-011 | ../src/main/java/com/gw/greatsoft/booking/AirAsia.java |
| SC-012 | ../src/main/java/com/gw/greatsoft/booking/DateParsingTest.java |
| SC-013 | ../src/main/java/com/gw/greatsoft/web/TiketController.java |
| SC-014 | ../src/main/java/com/gw/greatsoft/web/NegaraController.java |
| SC-015 | ../src/main/java/com/gw/greatsoft/web/RoleController.java |
| SC-016 | ../src/main/java/com/gw/greatsoft/web/KategoriPelangganController.java |
| SC-017 | ../src/main/java/com/gw/greatsoft/web/MenuGroupController.java |
| SC-018 | ../src/main/java/com/gw/greatsoft/web/AccGroupController.java |
| SC-019 | ../src/main/java/com/gw/greatsoft/web/CabangController.java |
| SC-020 | ../src/main/java/com/gw/greatsoft/web/BoderelLionController.java |
| SC-021 | ../src/main/java/com/gw/greatsoft/web/ArTagihanController.java |
| SC-022 | ../src/main/java/com/gw/greatsoft/web/BoderelGAInternationalController.java |
| SC-023 | ../src/main/java/com/gw/greatsoft/web/JenisPembayaranController.java |
| SC-024 | ../src/main/java/com/gw/greatsoft/web/ApplicationConfigController.java |
| SC-025 | ../src/main/java/com/gw/greatsoft/web/ArTagihanApproveController.java |
| SC-026 | ../src/main/java/com/gw/greatsoft/web/PermissionController.java |
| SC-027 | ../src/main/java/com/gw/greatsoft/web/BoderelKeretaApiController.java |
| SC-028 | ../src/main/java/com/gw/greatsoft/web/BahanController.java |
| SC-029 | ../src/main/java/com/gw/greatsoft/web/ItemBudgetingController.java |

| | |
|---|---|
| SC-030 | ../src/main/java/com/gw/greatsoft/web/ItemController.java |
| SC-031 | ../src/main/java/com/gw/greatsoft/web/JenisItemController.java |
| SC-032 | ../src/main/java/com/gw/greatsoft/web/UserController.java |
| SC-033 | ../src/main/java/com/gw/greatsoft/web/PekerjaController.java |
| SC-034 | ../src/main/java/com/gw/greatsoft/web/RelasiController.java |
| SC-035 | ../src/main/java/com/gw/greatsoft/web/BoderelAirAsiaController.java |
| SC-036 | ../src/main/java/com/gw/greatsoft/web/RelasiAkunController.java |
| SC-037 | ../src/main/java/com/gw/greatsoft/web/SatuanController.java |
| SC-038 | ../src/main/java/com/gw/greatsoft/web/BoderelSriwijayaController.java |
| SC-039 | ../src/main/java/com/gw/greatsoft/web/ProdukTemplateController.java |
| SC-040 | ../src/main/java/com/gw/greatsoft/web/ArInvoiceController.java |
| SC-041 | ../src/main/java/com/gw/greatsoft/web/ReportController.java |
| SC-042 | ../src/main/java/com/gw/greatsoft/web/ArInvoiceApproveController.java |
| SC-043 | ../src/main/java/com/gw/greatsoft/web/ArLppController.java |
| SC-044 | ../src/main/java/com/gw/greatsoft/web/PengajuanBkController.java |
| SC-045 | ../src/main/java/com/gw/greatsoft/web/PropinsiController.java |
| SC-046 | ../src/main/java/com/gw/greatsoft/web/BoderelCitilinkController.java |
| SC-047 | ../src/main/java/com/gw/greatsoft/web/BoderelController.java |
| SC-048 | ../src/main/java/com/gw/greatsoft/web/ArReceiptController.java |
| SC-049 | ../src/main/java/com/gw/greatsoft/web/RelasiGroupController.java |
| SC-050 | ../src/main/java/com/gw/greatsoft/web/AnggaranController.java |
| SC-051 | ../src/main/java/com/gw/greatsoft/web/AccArusKasController.java |
| SC-052 | ../src/main/java/com/gw/greatsoft/web/ItemAkunController.java |
| SC-053 | ../src/main/java/com/gw/greatsoft/web/AccCoaController.java |
| SC-054 | ../src/main/java/com/gw/greatsoft/web/MataUangController.java |
| SC-055 | ../src/main/java/com/gw/greatsoft/web/KotaController.java |
| SC-056 | ../src/main/java/com/gw/greatsoft/web/TestBacaGaruda.java |
| SC-057 | ../src/main/java/com/gw/greatsoft/web/AccJurnalController.java |
| SC-058 | ../src/main/java/com/gw/greatsoft/web/MenuController.java |
| SC-059 | ../src/main/java/com/gw/greatsoft/web/HomepageController.java |
| SC-060 | ../src/main/java/com/gw/greatsoft/HashCode.java |
| SC-061 | ../src/main/java/com/gw/greatsoft/view/CustomJasperReportsMultiFormatView.java |
| SC-062 | ../src/main/java/com/gw/greatsoft/model/NoFaktur.java |
| SC-063 | ../src/main/java/com/gw/greatsoft/model/ModelTagihanBayar.java |
| SC-064 | ../src/main/java/com/gw/greatsoft/model/BookingRequest.java |
| SC-065 | ../src/main/java/com/gw/greatsoft/model/Cookies.java |
| SC-066 | ../src/main/java/com/gw/greatsoft/model/ParamsLion.java |
| SC-067 | ../src/main/java/com/gw/greatsoft/model/TiketPenumpang.java |
| SC-068 | ../src/main/java/com/gw/greatsoft/model/FilterParam.java |
| SC-069 | ../src/main/java/com/gw/greatsoft/model/TiketRaw.java |
| SC-070 | ../src/main/java/com/gw/greatsoft/model/Tiket.java |
| SC-071 | ../src/main/java/com/gw/greatsoft/model/NamaSheet.java |
| SC-072 | ../src/main/java/com/gw/greatsoft/model/HistoriRequest.java |
| SC-073 | ../src/main/java/com/gw/greatsoft/WebMvcConfig.java |
| SC-074 | ../src/main/java/com/gw/greatsoft/security/AjaxAwareLoginUrlAuthenticationEntryPoint.java |
| SC-075 | ../src/main/java/com/gw/greatsoft/security/RestAuthenticationEntryPoint.java |
| SC-076 | ../src/main/java/com/gw/greatsoft/dao/PermissionDao.java |
| SC-077 | ../src/main/java/com/gw/greatsoft/dao/CabangDao.java |
| SC-078 | ../src/main/java/com/gw/greatsoft/dao/MenuGroupDao.java |
| SC-079 | ../src/main/java/com/gw/greatsoft/dao/RelasiAkunDao.java |

| | |
|---|---|
| SC-080 | ../src/main/java/com/gw/greatsoft/dao/ArInvoiceDetailDao.java |
| SC-081 | ../src/main/java/com/gw/greatsoft/dao/MataUangDao.java |
| SC-082 | ../src/main/java/com/gw/greatsoft/dao/ArTagihanApproveDao.java |
| SC-083 | ../src/main/java/com/gw/greatsoft/dao/PropinsiDao.java |
| SC-084 | ../src/main/java/com/gw/greatsoft/dao/UserDao.java |
| SC-085 | ../src/main/java/com/gw/greatsoft/dao/ArReceiptDao.java |
| SC-086 | ../src/main/java/com/gw/greatsoft/dao/KategoriPelangganDao.java |
| SC-087 | ../src/main/java/com/gw/greatsoft/dao/AccCoaDao.java |
| SC-088 | ../src/main/java/com/gw/greatsoft/dao/ItemAkunDao.java |
| SC-089 | ../src/main/java/com/gw/greatsoft/dao/AnggaranDao.java |
| SC-090 | ../src/main/java/com/gw/greatsoft/dao/RelasiGroupDao.java |
| SC-091 | ../src/main/java/com/gw/greatsoft/dao/RelasiDao.java |
| SC-092 | ../src/main/java/com/gw/greatsoft/dao/ProdukTemplateDao.java |
| SC-093 | ../src/main/java/com/gw/greatsoft/dao/ArLppDao.java |
| SC-094 | ../src/main/java/com/gw/greatsoft/dao/ItemDao.java |
| SC-095 | ../src/main/java/com/gw/greatsoft/dao/NegaraDao.java |
| SC-096 | ../src/main/java/com/gw/greatsoft/dao/AccJurnalDao.java |
| SC-097 | ../src/main/java/com/gw/greatsoft/dao/JenisPembayaranDao.java |
| SC-098 | ../src/main/java/com/gw/greatsoft/dao/AccGroupDao.java |
| SC-099 | ../src/main/java/com/gw/greatsoft/dao/ApplicationConfigDao.java |
| SC-100 | ../src/main/java/com/gw/greatsoft/dao/ItemBudgetingDao.java |
| SC-101 | ../src/main/java/com/gw/greatsoft/dao/BahanDao.java |
| SC-102 | ../src/main/java/com/gw/greatsoft/dao/JenisItemDao.java |
| SC-103 | ../src/main/java/com/gw/greatsoft/dao/jdbc/BoderelDao.java |
| SC-104 | ../src/main/java/com/gw/greatsoft/dao/jdbc/SystemDaoJdbc.java |
| SC-105 | ../src/main/java/com/gw/greatsoft/dao/jdbc/ItemDaoJdbc.java |
| SC-106 | ../src/main/java/com/gw/greatsoft/dao/jdbc/ArTagihanJdbc.java |
| SC-107 | ../src/main/java/com/gw/greatsoft/dao/jdbc/RelasiAkunDaoJdbc.java |
| SC-108 | ../src/main/java/com/gw/greatsoft/dao/jdbc/ArLppDaoJdbc.java |
| SC-109 | ../src/main/java/com/gw/greatsoft/dao/jdbc/ReportDao.java |
| SC-110 | ../src/main/java/com/gw/greatsoft/dao/jdbc/LookupDaoJdbc.java |
| SC-111 | ../src/main/java/com/gw/greatsoft/dao/jdbc/TiketDaoJdbc.java |
| SC-112 | ../src/main/java/com/gw/greatsoft/dao/jdbc/AccJurnalDaoJdbc.java |
| SC-113 | ../src/main/java/com/gw/greatsoft/dao/jdbc/RelasiDaoJdbc.java |
| SC-114 | ../src/main/java/com/gw/greatsoft/dao/jdbc/ItemAkunDaoJdbc.java |
| SC-115 | ../src/main/java/com/gw/greatsoft/dao/jdbc/ArTagihanApproveDaoJdbc.java |
| SC-116 | ../src/main/java/com/gw/greatsoft/dao/jdbc/ArInvoiceDaoJdbc.java |
| SC-117 | ../src/main/java/com/gw/greatsoft/dao/jdbc/MapResultSet.java |
| SC-118 | ../src/main/java/com/gw/greatsoft/dao/jdbc/BahanDaoJdbc.java |
| SC-119 | ../src/main/java/com/gw/greatsoft/dao/jdbc/ArInvoiceApproveDaoJdbc.java |
| SC-120 | ../src/main/java/com/gw/greatsoft/dao/jdbc/ArReceiptDaoJdbc.java |
| SC-121 | ../src/main/java/com/gw/greatsoft/dao/jdbc/PengajuanBkDaoJdbc.java |
| SC-122 | ../src/main/java/com/gw/greatsoft/dao/jdbc/MenuDaoJdbc.java |
| SC-123 | ../src/main/java/com/gw/greatsoft/dao/ArInvoiceDao.java |
| SC-124 | ../src/main/java/com/gw/greatsoft/dao/MenuDao.java |
| SC-125 | ../src/main/java/com/gw/greatsoft/dao/JenisJurnalDao.java |
| SC-126 | ../src/main/java/com/gw/greatsoft/dao/TiketDao.java |
| SC-127 | ../src/main/java/com/gw/greatsoft/dao/RoleDao.java |
| SC-128 | ../src/main/java/com/gw/greatsoft/dao/AirlineUserDao.java |
| SC-129 | ../src/main/java/com/gw/greatsoft/dao/ArInvoiceApproveDao.java |

| | |
|---|---|
| SC-130 | ../src/main/java/com/gw/greatsoft/dao/PengajuanBkDao.java |
| SC-131 | ../src/main/java/com/gw/greatsoft/dao/ArTagihanDao.java |
| SC-132 | ../src/main/java/com/gw/greatsoft/dao/AccTemplateDao.java |
| SC-133 | ../src/main/java/com/gw/greatsoft/dao/PekerjaDao.java |
| SC-134 | ../src/main/java/com/gw/greatsoft/dao/SatuanDao.java |
| SC-135 | ../src/main/java/com/gw/greatsoft/dao/KotaDao.java |
| SC-136 | ../src/main/java/com/gw/greatsoft/dao/AccArusKasDao.java |
| SC-137 | ../src/main/java/com/gw/greatsoft/hibernate/listener/AuditLogUsernameListener.java |
| SC-138 | ../src/main/java/com/gw/greatsoft/helper/FungsiUmum.java |
| SC-139 | ../src/main/java/com/gw/greatsoft/helper/TestBacaTiket.java |
| SC-140 | ../src/main/java/com/gw/greatsoft/helper/BacaLion.java |
| SC-141 | ../src/main/java/com/gw/greatsoft/helper/PesanConstraintStok.java |
| SC-142 | ../src/main/java/com/gw/greatsoft/helper/BacaCitilink.java |
| SC-143 | ../src/main/java/com/gw/greatsoft/helper/UrlPathHelperFixed.java |
| SC-144 | ../src/main/java/com/gw/greatsoft/helper/BacaSriwijaya.java |
| SC-145 | ../src/main/java/com/gw/greatsoft/interceptor/CorsInterceptor.java |
| SC-146 | ../src/main/java/com/gw/greatsoft/exception/ErrorInfo.java |
| SC-147 | ../src/main/java/com/gw/greatsoft/exception/GwException.java |
| SC-148 | ../src/main/java/com/gw/greatsoft/domain/ArInvoiceApprove.java |
| SC-149 | ../src/main/java/com/gw/greatsoft/domain/ProdukTemplate.java |
| SC-150 | ../src/main/java/com/gw/greatsoft/domain/JenisPembayaran.java |
| SC-151 | ../src/main/java/com/gw/greatsoft/domain/ApplicationConfig.java |
| SC-152 | ../src/main/java/com/gw/greatsoft/domain/AccJenisJurnal.java |
| SC-153 | ../src/main/java/com/gw/greatsoft/domain/AccJurnal.java |
| SC-154 | ../src/main/java/com/gw/greatsoft/domain/Menu.java |
| SC-155 | ../src/main/java/com/gw/greatsoft/domain/ApInvoice.java |
| SC-156 | ../src/main/java/com/gw/greatsoft/domain/Pekerja.java |
| SC-157 | ../src/main/java/com/gw/greatsoft/domain/Permission.java |
| SC-158 | ../src/main/java/com/gw/greatsoft/domain/ArTagihanDetail.java |
| SC-159 | ../src/main/java/com/gw/greatsoft/domain/Role.java |
| SC-160 | ../src/main/java/com/gw/greatsoft/domain/BahanDetail.java |
| SC-161 | ../src/main/java/com/gw/greatsoft/domain/AccJurnalDetail.java |
| SC-162 | ../src/main/java/com/gw/greatsoft/domain/TiketDetailPenumpang.java |
| SC-163 | ../src/main/java/com/gw/greatsoft/domain/PengajuanBkDetail.java |
| SC-164 | ../src/main/java/com/gw/greatsoft/domain/SettingPembayaran.java |
| SC-165 | ../src/main/java/com/gw/greatsoft/domain/Item.java |
| SC-166 | ../src/main/java/com/gw/greatsoft/domain/BaseEntity.java |
| SC-167 | ../src/main/java/com/gw/greatsoft/domain/TiketDetailRute.java |
| SC-168 | ../src/main/java/com/gw/greatsoft/domain/User.java |
| SC-169 | ../src/main/java/com/gw/greatsoft/domain/ArReceiptDetail.java |
| SC-170 | ../src/main/java/com/gw/greatsoft/domain/AccGroup.java |
| SC-171 | ../src/main/java/com/gw/greatsoft/domain/Boderel.java |
| SC-172 | ../src/main/java/com/gw/greatsoft/domain/RelasiAkun.java |
| SC-173 | ../src/main/java/com/gw/greatsoft/domain/Cabang.java |
| SC-174 | ../src/main/java/com/gw/greatsoft/domain/MenuGroup.java |
| SC-175 | ../src/main/java/com/gw/greatsoft/domain/MataUang.java |
| SC-176 | ../src/main/java/com/gw/greatsoft/domain/ArTagihan.java |
| SC-177 | ../src/main/java/com/gw/greatsoft/domain/Bahan.java |
| SC-178 | ../src/main/java/com/gw/greatsoft/domain/ArTagihanApprove.java |
| SC-179 | ../src/main/java/com/gw/greatsoft/domain/ArInvoiceDetail.java |

| | |
|---|---|
| SC-180 | ../src/main/java/com/gw/greatsoft/domain/AccArusKas.java |
| SC-181 | ../src/main/java/com/gw/greatsoft/domain/ArReceipt.java |
| SC-182 | ../src/main/java/com/gw/greatsoft/domain/AuditLogDenganUsername.java |
| SC-183 | ../src/main/java/com/gw/greatsoft/domain/ApInvoiceDetail.java |
| SC-184 | ../src/main/java/com/gw/greatsoft/domain/RelasiGroup.java |
| SC-185 | ../src/main/java/com/gw/greatsoft/domain/TiketDetailHarga.java |
| SC-186 | ../src/main/java/com/gw/greatsoft/domain/ArInvoice.java |
| SC-187 | ../src/main/java/com/gw/greatsoft/domain/AirlineUser.java |
| SC-188 | ../src/main/java/com/gw/greatsoft/domain/PengajuanBk.java |
| SC-189 | ../src/main/java/com/gw/greatsoft/domain/Propinsi.java |
| SC-190 | ../src/main/java/com/gw/greatsoft/domain/ItemAkun.java |
| SC-191 | ../src/main/java/com/gw/greatsoft/domain/Tiket.java |
| SC-192 | ../src/main/java/com/gw/greatsoft/domain/Kota.java |
| SC-193 | ../src/main/java/com/gw/greatsoft/domain/KategoriPelanggan.java |
| SC-194 | ../src/main/java/com/gw/greatsoft/domain/SettingNoFaktur.java |
| SC-195 | ../src/main/java/com/gw/greatsoft/domain/Satuan.java |
| SC-196 | ../src/main/java/com/gw/greatsoft/domain/Negara.java |
| SC-197 | ../src/main/java/com/gw/greatsoft/domain/ItemBudgeting.java |
| SC-198 | ../src/main/java/com/gw/greatsoft/domain/Anggaran.java |
| SC-199 | ../src/main/java/com/gw/greatsoft/domain/AccTemplate.java |
| SC-200 | ../src/main/java/com/gw/greatsoft/domain/ArLpp.java |
| SC-201 | ../src/main/java/com/gw/greatsoft/domain/AccSetting.java |
| SC-202 | ../src/main/java/com/gw/greatsoft/domain/JenisItem.java |
| SC-203 | ../src/main/java/com/gw/greatsoft/domain/Relasi.java |
| SC-204 | ../src/main/java/com/gw/greatsoft/domain/AccCoa.java |
| SC-205 | ../src/main/java/com/gw/greatsoft/domain/TransaksiEntity.java |
| SC-206 | ../src/main/java/com/gw/greatsoft/service/MonitoredService.java |
| SC-207 | ../src/main/java/com/gw/greatsoft/service/AppService.java |
| SC-208 | ../src/main/java/com/gw/greatsoft/service/impl/AppServiceImpl.java |
| SC-209 | ../src/main/java/com/gw/greatsoft/service/impl/ConfigServiceImpl.java |
| SC-210 | ../src/main/java/com/gw/greatsoft/service/impl/BaseEntityListener.java |
| SC-211 | ../src/main/java/com/gw/greatsoft/service/ConfigService.java |
| SC-212 | ../src/main/java/com/gw/greatsoft/service/AutowireHelper.java |
| SC-213 | ../src/main/java/com/gw/greatsoft/constant/TipePelanggan.java |
| SC-214 | ../src/main/java/com/gw/greatsoft/constant/TipeTagihan.java |
| SC-215 | ../src/main/java/com/gw/greatsoft/constant/ItemType.java |
| SC-216 | ../src/main/java/com/gw/greatsoft/constant/JenisApprove.java |
| SC-217 | ../src/main/java/com/gw/greatsoft/constant/ItemHistoriType.java |
| SC-218 | ../src/main/java/com/gw/greatsoft/constant/JenisTransaksi.java |
| SC-219 | ../src/main/java/com/gw/greatsoft/constant/TipePenumpang.java |
| SC-220 | ../src/main/java/com/gw/greatsoft/constant/ItemProfitType.java |
| SC-221 | ../src/main/java/com/gw/greatsoft/boderel/BodrelSriwijaya.java |
| SC-222 | ../src/main/java/com/gw/greatsoft/boderel/BoderelAirAsia.java |

Berdasarkan tabel cerita pengguna dan tabel kode sumber, pengembang malakukan identifikasi identifikasi hubungan kerunutan sehingga diidentifikasi terdapat 334 hubungan kerunutan, seperti terlihat pada tabel 4.24. Tabel hasil evaluasi precision recall *smartTravel* pada kolom ketiga.

Pada tabel tersebut juga dapat dilihat semakin kecil *similarity threshold* semakin banyak kerunutan yang teridentifikasi oleh system pada ambang batas 0.4 dan prosentase kata 0% system mengidentifikasi sebanyak 5379 kerunutan, dan yang benar sesuai dengan identifikasi pengembang hanya 279 dan terdapat 5100 sistem salah mengidentifikasi sehingga nilai presisi menjadi kecil. Dengan menambahkan prosentase jumlah kata yang terkandung dalam kode program dapat memperbaiki nilai presisi pada percobaan dapat dilihat pada tabel 4.24 dengan menambahkan prosentase antara 15% sampai 20% nilai presisi menjadi lebih baik.

Hampir sama dengan data sebelumnya, pemberian jumlah kandungan kata diatas 25% akan mengurangi jumlah kerunutan yang teridentifikasi, sehingga nilai presisipun menurun. Pada smartTravel nilai precision tertinggi berada pada threshold 0.5 dan jumlah kandungan kata 25% dengan nilai 0,217 dapat dilihat pada gambar

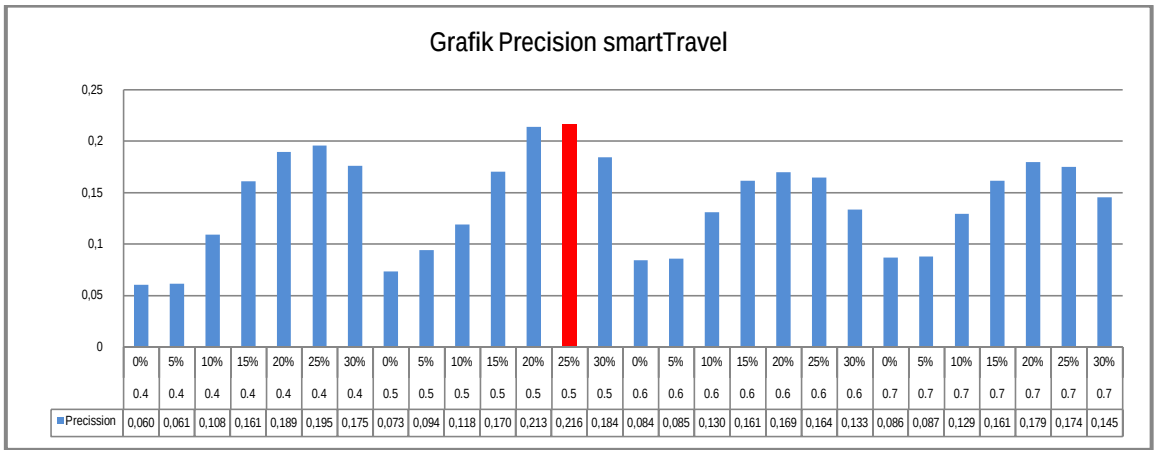
Tabel 4.24. hasil evaluasi *precision recall smartTravel*

| No | Similarity Threshold | Kandungan Jml Kata | Jml Identifikasi | | Jumlah Perbandingan | | | | Rata-Rata | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Pengembang | System | TP | FP | FN | TN | Precission | Recall |
| 1 | 0.4 | 0% | 334 | 5379 | 279 | 5100 | 55 | 12326 | 0.06009281 | 0.898731962 |
| 2 | 0.4 | 5% | 334 | 5135 | 273 | 4862 | 61 | 12564 | 0.061388222 | 0.878731962 |
| 3 | 0.4 | 10% | 334 | 2411 | 197 | 2214 | 137 | 15212 | 0.108664561 | 0.637450119 |
| 4 | 0.4 | 15% | 334 | 1196 | 157 | 1039 | 177 | 16387 | 0.160970508 | 0.491341298 |
| 5 | 0.4 | 20% | 334 | 663 | 114 | 549 | 220 | 16877 | 0.189201832 | 0.361248404 |
| 6 | 0.4 | 25% | 334 | 301 | 67 | 234 | 267 | 17192 | 0.195437462 | 0.218952714 |
| 7 | 0.4 | 30% | 334 | 203 | 53 | 150 | 281 | 17276 | 0.175766369 | 0.172970883 |
| 8 | 0.5 | 0% | 334 | 4705 | 251 | 4454 | 83 | 12972 | 0.073074636 | 0.850823482 |
| 9 | 0.5 | 5% | 334 | 1508 | 127 | 1381 | 73 | 5986 | 0.094046193 | 0.594680851 |
| 10 | 0.5 | 10% | 334 | 1868 | 170 | 1698 | 164 | 15728 | 0.118857305 | 0.579035062 |
| 11 | 0.5 | 15% | 334 | 889 | 130 | 759 | 204 | 16667 | 0.170266048 | 0.424634671 |
| 12 | 0.5 | 20% | 334 | 424 | 87 | 337 | 247 | 17089 | 0.213563392 | 0.312360764 |
| 13 | 0.5 | 25% | 334 | 182 | 53 | 129 | 281 | 17297 | 0.216592262 | 0.179721875 |
| 14 | 0.5 | 30% | 334 | 111 | 33 | 78 | 301 | 17348 | 0.184166667 | 0.113664287 |
| 15 | 0.6 | 0% | 334 | 3770 | 232 | 3538 | 102 | 13888 | 0.084270926 | 0.772603785 |
| 16 | 0.6 | 5% | 334 | 3583 | 225 | 3358 | 109 | 14068 | 0.085696786 | 0.746353785 |
| 17 | 0.6 | 10% | 334 | 1522 | 156 | 1366 | 178 | 16060 | 0.130817948 | 0.522595668 |
| 18 | 0.6 | 15% | 334 | 702 | 111 | 591 | 223 | 16835 | 0.161499077 | 0.352361944 |
| 19 | 0.6 | 20% | 334 | 310 | 58 | 252 | 276 | 17174 | 0.169720569 | 0.198004704 |
| 20 | 0.6 | 25% | 334 | 124 | 33 | 91 | 301 | 17335 | 0.164196429 | 0.107087357 |
| 21 | 0.6 | 30% | 334 | 80 | 23 | 57 | 311 | 17369 | 0.133541667 | 0.075627671 |
| 22 | 0.7 | 0% | 334 | 3625 | 230 | 3395 | 104 | 14031 | 0.086399672 | 0.764270452 |
| 23 | 0.7 | 5% | 334 | 3442 | 222 | 3220 | 112 | 14206 | 0.087896705 | 0.734895452 |
| 24 | 0.7 | 10% | 334 | 1434 | 152 | 1282 | 182 | 16144 | 0.129187402 | 0.502804002 |
| 25 | 0.7 | 15% | 334 | 642 | 103 | 539 | 231 | 16887 | 0.161242806 | 0.313820277 |
| 26 | 0.7 | 20% | 334 | 260 | 53 | 207 | 281 | 17219 | 0.179307463 | 0.173004704 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 27 | 0.7 | 25% | 334 | 106 | 30 | 76 | 304 | 17350 | 0.174886364 | 0.091462357 |
| 28 | 0.7 | 30% | 334 | 59 | 20 | 39 | 314 | 17387 | 0.14530303 | 0.063127671 |

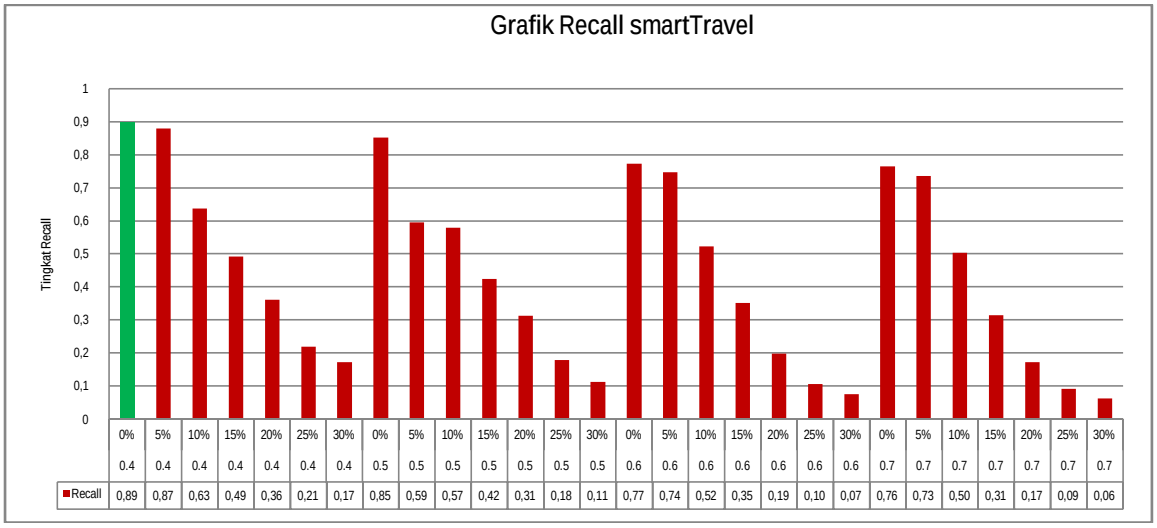
*Keterangan:*

| | |
|---|---|
| TP | True Positive |
| FP | False Positive |
| TN | True Negative |
| FN | False Negative |



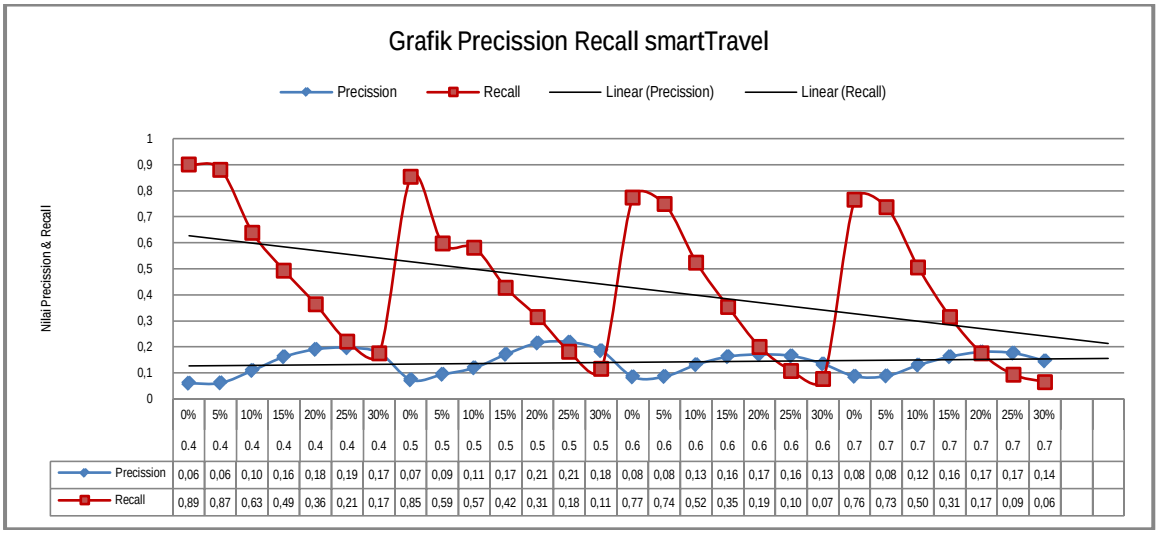

Gambar 4.19. Grafik Precision smartTravel

Gambar 4.20 menunjukkan kecenderungan yang sama untuk setiap ambang batas kemiripan kata apabila tidak ditambahkan jumlah kata yg terdapat dalam kode sumber. Nilai recall untuk prosentase jumlah kata 0% dan 5% berkisar antara 0.87 sampai 0.89. Setelah ditambahkan prosentase jumlah kata yang terkandung dalam kode sumber, terjadi penurunan pada nilai recall seiring dengan penambahan prosentase jumlah kata dalam kode sumber.



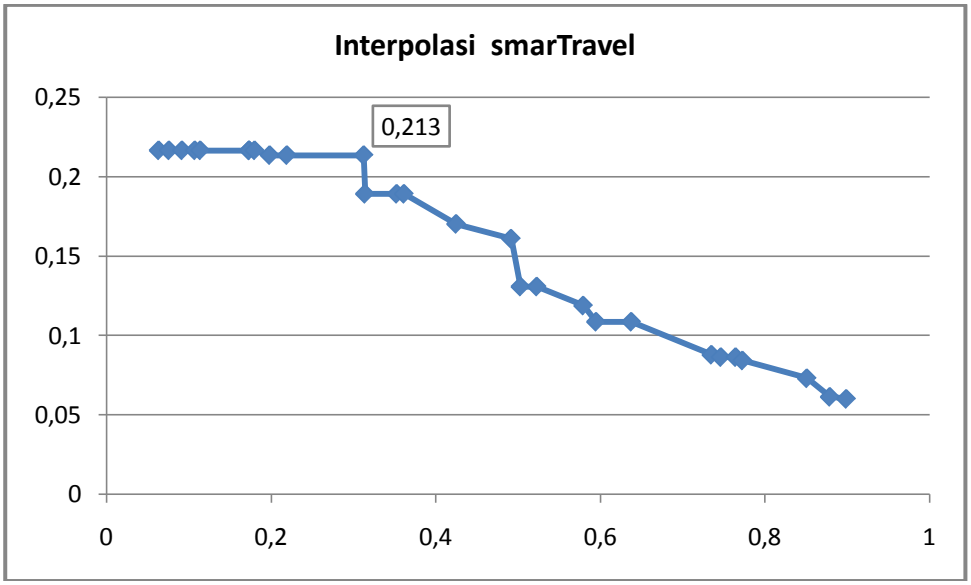

Gambar 4.20. Grafik Recall smartTravel

Garis *linier* pada Gambar 4.21 menunjukkan kecenderungan (*trend*) peningkatan nilai presisi yang meningkat seiring dengan peningkatan nilai ambang batas kemiripan, tetapi garis *linier* recall menunjukan kecenderungan menurun seiring dengan peningkatan nilai ambang batas kemiripan.




Gambar 4.21. Grafik Trend smartTravel

Berdasarkan gambar 4.22. Kurva interpolasi Precission dan Recall *smartTravel*, Perpotongan antara precission dan recall terletak pada titik prosentase jumlah kata 25% pada ambang batas kemiripan 0,5 merupakan titik terbaik dari presisi dan recall, dengan nilai 0,213 berada pada nilai recall 0,006 sampai 0,312.



Gambar 4.22. Interpolasi Precission dan Recall smartTravel

# BAB 5
# KESIMPULAN DAN SARAN

Bab ini menjelaskan tentang apa saja yang dapat disimpulkan dari penelitian yang sudah dilakukan. Kemudian, peneliti menjelaskan bagaimana saran atau kelanjutan penelitian pada masa mendatang.

## **5.1** Kesimpulan

Beberapa kesimpulan yang dapat ditarik dari hasil pengerjaan penelitian ini adalah sebagai berikut ini.

1. Dalam penelitian ini dikembangkan suatu metode untuk mengidentifikasi hubungan kerunutan antara cerita pengguna dengan kode sumber yang berbahasa Indonesia. Metode yang digunakan adalah menghubungkan kata-kata hasil ekstraksi dari cerita pengguna dan kode sumber menggunakan kemiripan kata dengan algoritma *trigram* yang terdapat dalam plugin postgreSQL. Hasil dari hubungan keduanya dioptimasi dengan menambahkan ambang batas perbandingan jumlah kata dalam kode sumber dengan jumlah kata dalam cerita pengguna.
2. Optimasi *parsing* kode sumber menggunakan aturan yang digunakan dalam penamaan kode sumber java atau konvensi penamaan java.
3. Hasil akhir uji coba oleh sistem dibandingkan dengan hasil identifikasi oleh pengembang adalah sebagai berikut:
   - Pada kasus pengujian *smartPortal* diperoleh nilai presisi 0,288 dengan nilai recall antara 0,556 sampai 0,226.
   - Pada kasus pengujian *smartAbsensi* diperoleh nilai presisi 0,285 berada pada nilai recall 0,008 sampai 0,336.
   - Pada kasus pengujian *smartTravel* diperoleh nilai presisi 0,213 berada pada nilai recall 0,006 sampai 0,312.

   Berdasarkan hasil dari ketiga percobaan tersebut maka nilai presisi tergolong kecil karena berada dibawah 0,5 hal ini disebabkan tidak ditemukannya kemiripan kata antara cerita pengguna dengan kode sumber yang masih menggunakan bahasa Inggris atau singkatan-singkatan.

## **5.2** Saran

Saran untuk penelitian mendatang adalah sebagai berikut ini.

1. Identifikasi kerunutan pada penelitian ini menggunakan pencocokan *string* sehingga tidak dapat mengidentifikasi berdasarkan makna kata. Pada penelitian mendatang perlu digunakan *wordnet* bahasa Indonesia agar presisi dari identifikasi sistem dapat lebih tinggi.
2. Pemrograman aplikasi telah berkembang dengan menggunakan *framework* berasitektur MVC (*model view controller*) yang memisahkan aplikasi yang digunakan untuk memanipulasi data, antarmuka dan kontrol. Dengan pemisahan ini biasanya terdapat perbedaan bahasa pemrograman antara antarmuka dengan yang lainnya. Pada penelitian ini hanya melakukan *parsing* pada kode sumber *java*, pada penelitian mendatang perlu dikembangkan hubungan kerunutan antara cerita pengguna dengan kode sumber *java*, HTML dan *javascript*.
3. Penelitian ini menggunakan cerita pengguna berbahasa Indonesia dan kode sumber *java* yang *identifier*nya berbahasa Indonesia, kendala utama bagi peneliti adalah mendapatkan dataset yang tepat, pada penelitian mendatang diharapkan terdapat dataset berupa cerita pengguna dan kode sumber berbahasa Indonesia sehingga antara penelitian yang satu dan yang lain dapat menggunakan dataset yang ideal serta bisa dibandingkan luarannya.

# DAFTAR PUSTAKA


Buckley, Jim, et al., (2003). “Towards a Taxonomy of Software Change.” *Journal of Software Maintenance and Evolution: Research and Practice, Vol. 17, hal. 309-332.*

Dorfman, Merlin dan Flynn, Richard F., (1984). “Arts-An Automated Requirements Traceabillity System.” *Journal of Systems and Software, Vol. 4, hal. 63-74.*

Héctor García, Eugenio Santos, Bruno Windels. (2008), "Traceabillity Management Architectures Supporting Total Traceabillity", *Varna, Bulgaria : i.Tech.*

Jelita Asian, Hugh E., Williams S.M.M. Tahaghoghi, (2005), “Stemming Indonesian”, *School of Computer Science and Information Technology RMIT University, GPO Box 2476V, Melbourne 3001, Australia.*

Juergen Rilling, René Witte,Yonggang Zhang, (2007), "Automatic Traceabillity Recovery: An Ontological Approach", *Lexington, KY, USA. : s.n., , Vols. March 22-23. ACM ISBN 1-59593-6017/03/07.*

Kannenberg, Andy dan Saiedian, Hossein, (2009). “Why Software Requirements Traceabillity Remains a Challenge.” *The Journal of Defense Software Engineering, Juli, Vol. 22, hal. 14-19.*

Kenneth E. Kendall & Julie E. Kendall, (2011). "System Analysis and Design", *Prentice Hall*

Mark Grechanik, Kathryn S. McKinley, Dewayne E. Perry, (2007), "Recovering And Using Use-Case-Diagram-To-Source-Code Traceabillity Links",, *Cavtat near Dubrovnik, Croatia : s.n., 2007, Vols. 3-7. ACM 978-1-59593-811-4/07/0009.*

Mens, Tom dan Demeyer, Serge, (2008), “Software Evolution.” *Berlin : Springer.*

P. Lago, H. Muccini, and H. van Vliet, (2009). "A Scoped Approach To Traceabillity Management", *Journal of Systems and Software, 82(1):168 - 182, Special Issue: Software Performance - Modeling and Analysis.*

Rilling, dkk.( 2007), "Automatic Traceabillity Recovery An Ontological Approach" *Dept. of Computer science and SE Montreal, QC H3G1M8, Canada.*

Rochimah, Siti, Wan Kadir, W. M. N. dan Abdullah, Abdul H, (2007). “An Evaluation of Kerunutan Approaches to Support Software Evolution.” *International Conference on Software Engineering Advances, 2007.*

Sun Microsystem, (1995-1999) , “9 - Naming Conventions”, http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-135099.html

The PostgreSQL Global Development Group, (1996-2016), “PostgreSQL 9.4.8 *Documentation”, https://www.postgresql.org/docs/9.4/static/pgtrgm.html*

# BIOGRAFI PENULIS

Penulis dilahirkan di Mojokerto pada tanggal 24 Juni 1974, merupakan anak pertama dari dua bersaudara. Penulis telah menempuh pendidikan SLTA di SMA Negeri Sooko Mojokerto. Pada tahun 1992 penulis melanjutkan pendidikan ke jenjang pendidikan tinggi di S-1 Teknik Informatika Universitas Dr. Soetomo dan lulus tahun 1998. Pada tahun 2001 penulis mendapatkan amanah untuk menjadi tenaga pengajar di Universitas Dr. Soetomo di Program Studi Teknik Informatika. Kemudian, tahun 2011 penulis melanjutkan studi S-2 di Institut Teknologi Sepuluh Nopember di Program Studi Teknik Informatika dan lulus sebagai Magister Komputer pada tahun 2016. Untuk korespondensi, penulis dapat dihubungi melalui email hengki.suhartoyo@gmail.com.